

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

«До захисту допущено»

Науковий керівник кафедри

_____ І.А. Дичка

«__»_____2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напрямку підготовки 6.050103 «Програмна інженерія»

**на тему: «Програмне забезпечення для підтримки дидактичної
діяльності викладачів молодших класів загальноосвітньої школи»**

Виконала:

Студентка IV курсу, групи КП-52

Левчук Ольга Сергіївна _____

Керівник:

Доцент кафедри ПЗКС, к.т.н., доцент,

Заболотня Т.М. _____

Консультант з нормоконтролю:

Доцент кафедри ПЗКС, к.т.н.,

Онай М.В. _____

Рецензент:

Доцент кафедри ММСА, к.т.н., доцент,

Дідковська М.В. _____

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студентка _____

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) –
6.050103 «Програмна інженерія» («Програмне забезпечення систем»)

ЗАТВЕРДЖУЮ

Науковий керівник кафедри

_____ І.А. Дичка

«__» _____ 2018 р.

ЗАВДАННЯ
на дипломний проект студенту
Левчук Ользі Сергіївні

1. Тема проекту «Програмне забезпечення для підтримки дидактичної діяльності викладачів молодших класів загальноосвітньої школи», керівник проекту Заболотня Тетяна Миколаївна, к.т.н., доцент, затверджені наказом по університету від «22» травня 2019. № 1331-С
2. Термін подання студентом проекту «18» червня 2018 р.
3. Вихідні дані до проекту: див. Технічне завдання.
4. Зміст пояснювальної записки:
 - аналіз існуючих програмних рішень;
 - обґрунтування вибору засобів реалізації;
 - структурно-алгоритмічна організація;
 - аналіз розроблених програмних засобів.
5. Перелік обов'язкового графічного матеріалу:
 - блок-схема алгоритму роботи модулю підтримки соціально-психологічних процесів розвитку дитини (креслення);
 - блок-схема алгоритму роботи модулю підтримки гностичних процесів розвитку дитини (креслення);

- блок-схема алгоритму роботи модулю підтримки інструментальних процесів розвитку дитини (плакат);
- інтерфейс програмних модулів (плакат).

6. Консультанти розділів проекту

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| Нормоконтроль | Онай М.В., старший викладач | | |

7. Дата видачі завдання «05» грудня 2018 р.

Календарний план

| № з/п | Назва етапів виконання дипломного проекту | Термін виконання етапів проекту | Примітка |
|-------|---|---------------------------------|----------|
| 1. | Вивчення літератури за тематикою проекту | 14.12.2018 | |
| 2. | Розроблення та узгодження технічного завдання | 28.12.2018 | |
| 3. | Розроблення структури програмного додатку | 10.01.2019 | |
| 4. | Підготовка матеріалів першого розділу дипломного проекту | 21.01.2019 | |
| 6. | Підготовка матеріалів другого розділу дипломного проекту | 30.01.2019 | |
| 7. | Програмна реалізація програмного додатку | 20.02.2019 | |
| 8. | Тестування програмного додатку | 20.04.2019 | |
| 9. | Підготовка матеріалів третього розділу дипломного проекту | 25.04.2019 | |
| 10. | Підготовка матеріалів четвертого розділу дипломного проекту | 07.05.2019 | |
| 11. | Підготовка графічної частини дипломного проекту | 20.05.2019 | |
| 12. | Оформлення документації дипломного проекту | 01.06.2019 | |

Студент

О. С. Левчук

Керівник проекту

Т. М. Заболотня

АНОТАЦІЯ

Даний дипломний проект присвячений розробленню програмного забезпечення для підтримки дидактичної діяльності викладачів молодших класів загальноосвітньої школи.

Розроблене програмне забезпечення є ігровим програмним комплексом, що включає в себе не тільки поліпшені аналоги існуючих ігор, що виконують гностичну і інструментальну функції, але й ігри, які виконують соціально-психологічну функцію, зокрема, націлені на навчання базової ввічливості. Саме така форма додатку обрана для того, щоб забезпечити максимальну незалежність його роботи від зовнішніх факторів, наприклад, від підключення до Інтернет. Програма є достатньо легковісною, щоб батьки або вчителі мали змогу запустити її на майже будь-якому комп'ютері або ноутбукі. Єдине існуюче обмеження – операційна система має бути тільки Windows, але у майбутньому цей недолік буде виправлений, оскільки додаток реалізується на основі багатоплатформового інструменту для розроблення двовимірних та тривимірних додатків Unity.

Також система є легко розширюваною, оскільки кожна з ігор, що входить до складу програми, являє собою окрему ігрову сцену, що робить взаємозв'язок між іграми мінімальним. Ігрові сцени містять об'єкти, які будуть використовуватись у грі і можуть служити як для створення меню гри, так і для моделювання ігрового рівня.

ABSTRACT

This diploma project is devoted to the development of software to support the teaching staff of junior high school teachers.

The developed software is a gaming software complex, which includes not only improved analogs of existing games that perform gnostic and instrumental functions but also games that perform a socio-psychological function, in particular, aimed at learning basic courtesy. The exact form of the application was chosen to ensure the maximum independence of its work from other factors, for example, from connecting to the Internet. The program is light enough to allow parents or teachers to run it on almost any computer or laptop. The only current limitation is the operating system that should be Windows, but in the future, it would be fixed as the application is implemented based on Unity - a multi-platform tool for the development of two-dimensional and three-dimensional applications.

Also, the system is easily expandable, because each of the games included in the program is a separate game scene, which makes the interplay between games minimal. The game scenes contain objects that are used in the game and can serve both to create a game menu and to simulate the game level.

ДП.045480-01-90 Програмне забезпечення для підтримки дидактичної діяльності викладачів молодших класів загальноосвітньої школи. Відомість проекту

| Позначення | Найменування | Кіл-ть | Примітка |
|-----------------|---------------------------|--------|----------|
| | Документація проекту | | |
| | | | |
| ДП.045480-02-91 | Програмне забезпечення | 5 | |
| | для підтримки дидактичної | | |
| | діяльності викладачів | | |
| | молодших класів | | |
| | загальноосвітньої школи. | | |
| | Технічне завдання | | |
| | | | |
| ДП.045480-03-81 | Програмне забезпечення | 51 | |
| | для підтримки дидактичної | | |
| | діяльності викладачів | | |
| | молодших класів | | |
| | загальноосвітньої школи. | | |
| | Пояснювальна записка | | |
| | | | |
| ДП.045480-04-51 | Програмне забезпечення | 4 | |
| | для підтримки дидактичної | | |
| | діяльності викладачів | | |
| | молодших класів | | |
| | загальноосвітньої школи. | | |
| | Програма та методика | | |
| | тестування | | |
| | | | |
| | | | |
| | | | |

| Позначення | Найменування | Кіл-ть | Примітка |
|-----------------|---------------------------|--------|----------|
| | | | |
| ДП.045480-05-34 | Програмне забезпечення | 9 | |
| | для підтримки дидактичної | | |
| | діяльності викладачів | | |
| | молодших класів | | |
| | загальноосвітньої школи. | | |
| | Керівництво користувача | | |
| | | | |
| ДП.045480-06-99 | Програмне забезпечення | 1 | |
| | для підтримки дидактичної | | |
| | діяльності викладачів | | |
| | молодших класів | | |
| | загальноосвітньої школи. | | |
| | Модуль підтримки | | |
| | соціально-психологічних | | |
| | процесів розвитку дитини. | | |
| | Блок-схема алгоритму | | |
| | | | |
| ДП.045480-07-99 | Програмне забезпечення | 1 | |
| | для підтримки дидактичної | | |
| | діяльності викладачів | | |
| | молодших класів | | |
| | загальноосвітньої школи. | | |
| | Модуль підтримки | | |
| | гностичних процесів | | |
| | розвитку дитини. | | |
| | Блок-схема алгоритму | | |
| | | | |

[illegible]

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Науковий керівник кафедри

_____ І.А. Дичка

“ ____ ” _____ 2018 р.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ ДИДАКТИЧНОЇ
ДІЯЛЬНОСТІ ВИКЛАДАЧІВ МОЛОДШИХ КЛАСІВ
ЗАГАЛЬНООСВІТНЬОЇ ШКОЛИ

Технічне завдання

ДП.045490-02-91

“ПОГОДЖЕНО”

Керівник проекту:

_____ Т.М. Заболотня

Нормоконтроль:

_____ М.В. Онай

Виконавець:

_____ О.С. Левчук

ЗМІСТ

| | |
|---|---|
| 1. Найменування та галузь застосування..... | 3 |
| 2. Підстава для розроблення | 3 |
| 3. Призначення розробки..... | 3 |
| 4. Вимоги до програмного продукту..... | 3 |
| 5. Вимоги до проектної документації | 4 |
| 6. Етапи проектування | 5 |
| 7. Порядок тестування розробки | 5 |

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Програмне забезпечення для підтримки дидактичної діяльності викладачів молодших класів загальноосвітньої школи.

Галузь застосування: інформаційні технології.

2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ

Підставою для розроблення є завдання на дипломне проектування, затверджене кафедрою програмного забезпечення комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім. Ігоря Сікорського).

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для використання в якості засобу для постановки та перевірки завдань, що націлені на тренування різних категорій навчальних процесів дітей молодших класів.

4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

Програмне забезпечення повинно забезпечувати такі основні функції:

1. можливість автоматично формулювати та перевіряти завдання, націлені на підтримку соціально-психологічних процесів розвитку дитини;
2. проведення аналізу правильності голосової відповіді за допомогою технології розпізнавання мови Google Speech-to-Text;
3. можливість автоматично формулювати та перевіряти завдання, націлені на підтримку гностичних процесів розвитку дитини;

4. можливість автоматично формувати та перевіряти завдання, націлені на підтримку інструментальних процесів розвитку дитини;
5. можливість використання додатку на різних операційних системах.

Розробку виконати за допомогою ігрового рушія Unity3D.

Додаткові вимоги:

1. дизайн інтерфейсу відповідно до типу поточної вправи;
2. наявність головного меню для навігації між вправами;
3. забезпечення надійності використання системи.

5. ВИМОГИ ДО ПРОЕКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання проекту повинна бути розроблена наступна документація:

- 1) пояснювальна записка;
- 2) програма та методика тестування;
- 3) керівництво користувача;
- 4) креслення:
 - «Модуль підтримки соціально-психологічних процесів розвитку дитини. Блок-схема алгоритму»;
 - «Модуль підтримки гностичних процесів розвитку дитини. Блок-схема алгоритму».

6. ЕТАПИ ПРОЕКТУВАННЯ

| | |
|---|------------|
| Вивчення літератури за тематикою роботи..... | 14.11.2018 |
| Розроблення та узгодження технічного завдання | 28.11.2018 |
| Розроблення структури програмного забезпечення..... | 15.12.2018 |
| Розроблення дизайну програмних модулів | 03.02.2019 |
| Програмна реалізація програмного забезпечення | 17.03.2019 |
| Тестування програмного забезпечення..... | 03.04.2019 |
| Підготовка матеріалів текстової частини проекту | 28.04.2019 |
| Підготовка матеріалів графічної частини проекту | 12.05.2019 |
| Оформлення технічної документації проекту..... | 25.05.2019 |

7. ПОРЯДОК ТЕСТУВАННЯ РОЗРОБКИ

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ І.А. Дичка

“ ” _____ 2019р.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ ДИДАКТИЧНОЇ
ДІЯЛЬНОСТІ ВИКЛАДАЧІВ МОЛОДШИХ КЛАСІВ
ЗАГАЛЬНООСВІТНЬОЇ ШКОЛИ

Пояснювальна записка

ДП.045480-03-81

“ПОГОДЖЕНО”

Керівник проекту:

_____ Т.М. Заболотня

Нормоконтроль:

_____ М.В. Онай

Виконавець:

_____ О. С. Левчук

2019

ЗМІСТ

| | |
|--|----|
| ВСТУП | 4 |
| 1. АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ | 6 |
| 1.1. Перелік вимог до програмного забезпечення..... | 6 |
| 1.2. Аналіз характеристик існуючих програмних рішень | 7 |
| 1.3. Результати аналізу | 12 |
| 2. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ..... | 14 |
| 2.1. Вибір ігрового рушія для розроблення | 14 |
| 2.2. Вибір API для використання технології розпізнавання мови..... | 18 |
| 3. СТРУКТУРНО-АЛГОРИТМІЧНА ОРГАНІЗАЦІЯ | 21 |
| 3.1. Аналіз вимог до програмних засобів..... | 21 |
| 3.2. Модуль підтримки соціально-психологічних процесів розвитку дитини..... | 26 |
| 3.3. Модуль підтримки гностичних процесів розвитку дитини | 30 |
| 3.4. Модуль підтримки інструментальних процесів розвитку дитини ... | 32 |
| 4. АНАЛІЗ РОЗРОБЛЕНИХ ПРОГРАМНИХ ЗАСОБІВ | 35 |
| 4.1. Особливості реалізації | 35 |
| 4.2. Інтерфейс та вміст модулів..... | 37 |
| 4.3. Тестування програмного забезпечення..... | 42 |
| 4.4. Рекомендації щодо подальшого вдосконалення..... | 44 |
| ВИСНОВКИ..... | 46 |
| СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ..... | 48 |
| ДОДАТКИ..... | 51 |

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

Ігровий рушій – інструмент для розроблення ігрових додатків. Важливою характеристикою рушія є можливість створення багатоплатформових ігор;

Кросплатформеність – властивість програмного забезпечення працювати більш, ніж на одній програмній чи апаратній платформах;

Спрайт – 2D зображення, яке використовується в комп'ютерній графіці;

UI – користувацький інтерфейс;

UX – почуття людини при користуванні будь-яким продуктом або системою;

ВСТУП

Процес автоматизації різних сфер людської діяльності стає все більш значущим у сучасному суспільстві. Зокрема, важко переоцінити вплив застосування комп'ютерних технологій в освітній галузі, адже розроблення та впровадження програмного забезпечення з психолого-педагогічною метою відчутно сприяє удосконаленню форм та методів навчального процесу.

Початкова школа є одним з найважливіших етапів у розвитку дитини, бо саме на цьому етапі починають своє формування майже всі основні вміння та навички людини. Тому вдала організація навчального процесу молодших школярів має велике значення для ефективності засвоєння матеріалу та набуття ними необхідних компетенцій.

Основна частина навчання дітей часто проводиться вчителями в ігровій формі, оскільки це найбільш ефективна методика для підтримки розсіяної уваги дитини. А у зв'язку зі стрімким зростанням популярності ігрової індустрії і тим, що віковий поріг, необхідний для користування комп'ютером, зменшується, впровадження навчальних комп'ютерних ігор допомогло б викладачам в утриманні уваги дитини. Тому при розробленні програмного забезпечення доцільно приділяти увагу саме вивченню способів комп'ютерної гейміфікації навчального процесу школярів.

Слід зауважити, що на даний момент не існує повноцінного програмного забезпечення, яке включало б вправи для розвитку учнів молодших класів в декількох напрямках, окрім деяких стаціонарних ігрових приладів, наприклад інтерактивних столів. Слід зауважити, що навіть такий прилад, який є досить дорогим, не містить ігор, які виконують соціально-психологічну функцію у розвитку дитини. Саме тому перед батьками та вчителями стає складне завдання щодо підбору набору навчального програмного забезпечення і правильного розподілу часу на роботу з кожним з них. Таким чином, актуальною є задача розроблення ігрового

комплексу для охоплення максимальної кількості категорій розвитку дитини.

1. АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ

1.1. Перелік вимог до програмного забезпечення

Для коректного вибору характеристик існуючих програмних рішень, на які в першу чергу буде звернута увага при аналізі, а також для формування множини критеріїв оцінювання їх ефективності, визначимо основну мету розроблення даного програмного забезпечення. Головна мета програмного продукту, створюваного в рамках даного дипломного проекту – забезпечення підтримки розвитку дитини з точки зору всіх існуючих категорій розвиваючих процесів. У літературі виділяють наступні категорії розвиваючих процесів:

- інструментальна: націлена на формування основних навичок та вмінь;
- гностична: націлена на формування знань та розвиток мислення;
- соціально-психологічна: націлена на розвиток комунікативних навичок [1].

Існуючі сьогодні різноманітні дитячі розвиваючі десктопні ігри часто сприяють розвитку дітей тільки в одному напрямку, або мають не дуже привабливий інтерфейс, що може значно знизити увагу та зацікавленість дитини грою. До того ж, необхідно враховувати апаратну складову, чи є можливим використання продукту без купівлі додаткового обладнання, тому основними характеристиками, за якими будемо оцінювати вже існуючі програмні засоби, є:

- інтерфейс;
- кількість категорій розвиваючих процесів, на які впливає ПЗ;
- універсальність використання;
- автономність.

Зупинимось більш детально на інтерфейсі та уточнимо, з якої точки зору він буде розглядатися в рамках даного дипломного проекту.

1.1.1. Вимоги до інтерфейсу

В залежності від типу гри вимоги до інтерфейсу мають бути різними:

- якщо гра впливає на інструментальні процеси розвитку дитини – кількість елементів на екрані має бути мінімальною, щоб забезпечити концентрування уваги дитини лише на вправі, бо основною метою є формування основних навичок та вмінь;
- якщо гра впливає на гностичні процеси розвитку дитини, основною вимогою до інтерфейсу є привабливість, оскільки ігри даного типу розвивають мислення і у своїй більшості базуються на використанні зображень. Якщо головними засобами для розвитку мислення не є зображення, то додатковою умовою буде невелика перевантаженість інтерфейсу, щоб сконцентрувати увагу дитини на головному засобі для тренування гностичних категорій розвитку дитини;
- якщо гра впливає на соціально-психологічні процеси розвитку дитини – інтерфейс має бути привабливим та не перевантаженим, оскільки основна увага дитини має бути націлена тільки на саму вправу.

Для поширення аудиторії необхідно, щоб батькам дитини не потрібно було купувати додаткове обладнання, тому головною характеристикою універсальності програмного засобу можливість запуску програми на комп'ютері або ноутбучі. Поняття автономності в даному контексті буде означати можливість виконувати вправи, запропоновані програмним засобом без третьої особи, яка буде перевіряти коректність відповіді.

1.2. Аналіз характеристик існуючих програмних рішень

1.2.1. "Учимся читать. Игры для обучения чтению"

"Учимся читать. Игры для обучения чтению" [2] – російськомовна гра, що дозволяє вчитись читати, має багато рівнів, які стають складнішими і складнішими: спочатку букви, потім склади, а потім речення, до того ж гра

має звуковий супровід. ПЗ впливає на інструментальні процеси розвитку (знань та вмінь дитини), має не перевантажений інтерфейс, тому можна стверджувати, що даний комп'ютерний додаток відповідає поставленим вище вимогам до інтерфейсу. Крім того, гра є автономною, а перевірка коректності відповіді дитини виконується в програмний спосіб. Також програма задовольняє вимогу універсальності, оскільки може запускатися на комп'ютері або ноутбукі. Мінімальні системні вимоги даного комп'ютерного додатку: ОС Windows XP/Vista/7/8/10, Pentium 200 MHz, 64 Мб ОЗУ.



Рис. 1.1. Скріншот з гри "Учимся читать. Игры для обучения чтению"

1.2.2. «Дракоша и Семь чудес света»

«Дракоша и Семь чудес света» [3] – пізнавальна гра, яка дозволяє вивчити історію в ігровій формі. Впливає тільки на гностичні процеси розвитку дитини, отже, так само як і гра, описана в п.п.1.2.1, має тільки третину бажаних функцій. Інтерфейс є привабливим, але основним засобом для впливу на гностичні процеси розвитку дитини є текст, а інтерфейс містить дуже багато різних намальованих деталей, що суперечить завданню сконцентрувати увагу дитини на вправі (рис. 1.2), тобто інтерфейс не є

цілком підходящим для даного типу гри. Гра є автономною, перевірка коректності відповіді дитини виконується програмно. Задовольняє вимогу універсальності, оскільки даний програмний засіб може бути запущеним на комп'ютері або ноутбучі. Мінімальні системні вимоги даного комп'ютерного додатку: Windows 98/Me/2000/XP, Pentium 200 MHz, 64 Мб ОЗУ.

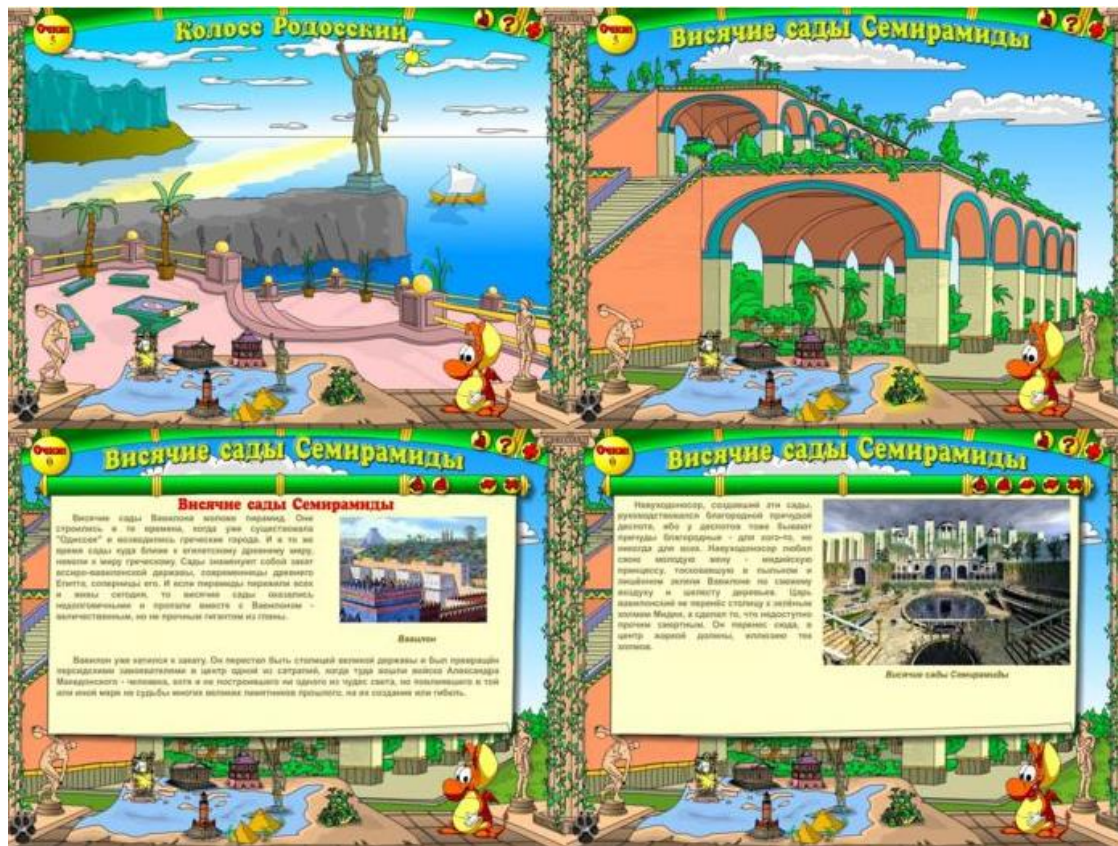


Рис. 1.2. Скріншот з гри «Дракоша и Семь чудес света»

1.2.3. «Веселые игры для развития речи и слуха»

«Веселые игры для развития речи и слуха» [4] – гра соціально-психологічного типу, має не перевантажений, але привабливий інтерфейс (див. рис. 1.3). Не є автономною, оскільки для перевірки коректності відповіді потрібна участь третьої особи, яка, згідно з рекомендацією програми, має бути старша за дитину. Задовольняє вимогу універсальності, оскільки працює на комп'ютері або ноутбучі. Системні вимоги даного

комп'ютерного додатку: Windows 2000/XP/Vista, Pentium III 500 МГц, 64 МБ ОЗУ.



Рис. 1.3. Скріншот з гри «Веселые игры для развития речи и слуха»

1.2.4. «Губка Боб Квадратные Штаны учит печатать»

«Губка Боб Квадратные Штаны учит печатать»[5] – гра з дуже приємним UX та гарним UI (див. рис 1.4), який вчить дітей методу сліпого друку, тобто впливає тільки на інструментальні процеси розвитку дитини. До того ж, вона є дуже популярною серед дітей, оскільки носить назву та використовує ілюстрації з популярного мультсеріалу. Є автономною, перевірка коректності відповіді дитини виконується програмно. Задовольняє вимогу універсальності, оскільки може працювати на комп'ютері або ноутбучі. Мінімальні системні вимоги даного комп'ютерного додатку: Windows 2000 SP4/XP SP1; Pentium II 350 МГц; 256 Мб ОЗУ.



Рис. 1.4. Скріншот з гри «Губка Боб Квадратные Штаны учит печатать»

1.2.5. Інтерактивні столи

На сьогоднішній день існує багато варіантів ігрових столів, які являють собою сенсорний стіл з вбудованим програмним забезпеченням, яке впливає на розвиток дидактичних та інструментальних процесів розвитку дитини.

Як приклад, можна навести інтерактивний стіл Kids Table 32 [6] (див. рис. 1.5), в якому міститься близько 200 різноманітних ігор, жодна з яких не підтримує соціально-психологічні процеси розвитку дитини, але кожна з яких має дуже яскравий і водночас мінімалістичний інтерфейс, що робить даний прилад дуже привабливим для дитини. Ігровий стіл є автономним, перевірка коректності відповіді дитини виконується програмними засобами. Найбільшим недоліком такої системи є її ціна, оскільки вона може

функціонувати лише в парі з конкретною апаратурою. Таким чином, дана програма не задовольняє вимогу універсальності.



Рис. 1.5. Інтерактивний стіл Kids Table 32

1.3. Результати аналізу

У табл. 1.1 та 1.2. систематизовано відповідність досліджуваних програм заданим критеріям оцінювання.

Таблица 1.1

Порівняння інтерфейсів та кількості підтримуваних категорій процесів розвитку дитини існуючого ПЗ

| Назва | Інтерфейс | Кількість категорій (з 3) |
|---|-----------|------------------------------|
| "Учимся читать. Игры для обучения чтению" | + | 1 |
| Дракоша и Семь чудес света | — | 1 |
| Веселые игры для развития речи и слуха | + | 1 |
| Губка Боб Квадратные Штаны учит печатать | + | 1 |

Продовження табл. 1.1

| Назва | Інтерфейс | Кількість категорій (з 3) |
|--------------------|-----------|------------------------------|
| Інтерактивні столи | + | 2 |

Таблиця 1.2

Порівняння універсальності та автономності існуючого ПЗ

| Назва | Універсальність | Автономність |
|---|-----------------|--------------|
| "Учимся читать. Игры для обучения чтению" | + | + |
| Дракоша и Семь чудес света | + | + |
| Веселые игры для развития речи и слуха | + | — |
| Губка Боб Квадратные Штаны учит печатать | + | + |
| Інтерактивні столи | — | + |

Виходячи з огляду існуючих рішень можна зазначити, що жодне з них повністю не відповідає поставленим вимогам, оскільки жодне з них не впливає на всі три категорії процесів розвитку дитини, найбільш відповідним до вимог є категорія інтерактивних столів, але потребує купівлі додаткового обладнання.

2. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ

На сьогоднішній день існує багато програмного забезпечення, що допомагає у розробленні різних видів проектів, у тому числі ігрового характеру. Інструментарій, спрямований на створення ігрового додатку, називають ігровим рушієм.

2.1. Вибір ігрового рушія для розроблення

Розглянемо види найпопулярніших ігрових рушіїв та мов програмування, що використовуються для створення проекту.

Unity3D [7] – інструмент для розроблення кросплатформних додатків. Дозволяє розробляти проекти не у форматах 2D та 3D, має підтримку найпопулярніших графічних бібліотек, таких як DirectX [8] та OpenGL [9]. Розробникам надається вибір з двох мов програмування для створення власного проекту: C# та JavaScript:

- *C#* [10] – об'єктно-орієнтована мова програмування, яка була спроектована для платформи .NET. Має достатньо багато стандартних бібліотек, що спрощує розроблення, але є оптимальною лише для розроблення не дуже великих проектів.
- *JavaScript* [11] – динамічна та об'єктно-орієнтована мова програмування. Є найпоширенішою мовою програмування для створення веб-сторінок або браузерних ігор.

Серед переваг цього інструменту слід зазначити, що за його допомогою можна створити достатньо легковісний проект у порівнянні з іншими ігровими рушіями, оскільки він включає в себе невеликий набір інструментів для розроблення, завдяки чому має інтуїтивно зрозумілий інтерфейс (див. рис. 2.1). До того ж він є достатньо стабільним, оскільки має постійну підтримку у вигляді регулярного випуску нових версій.

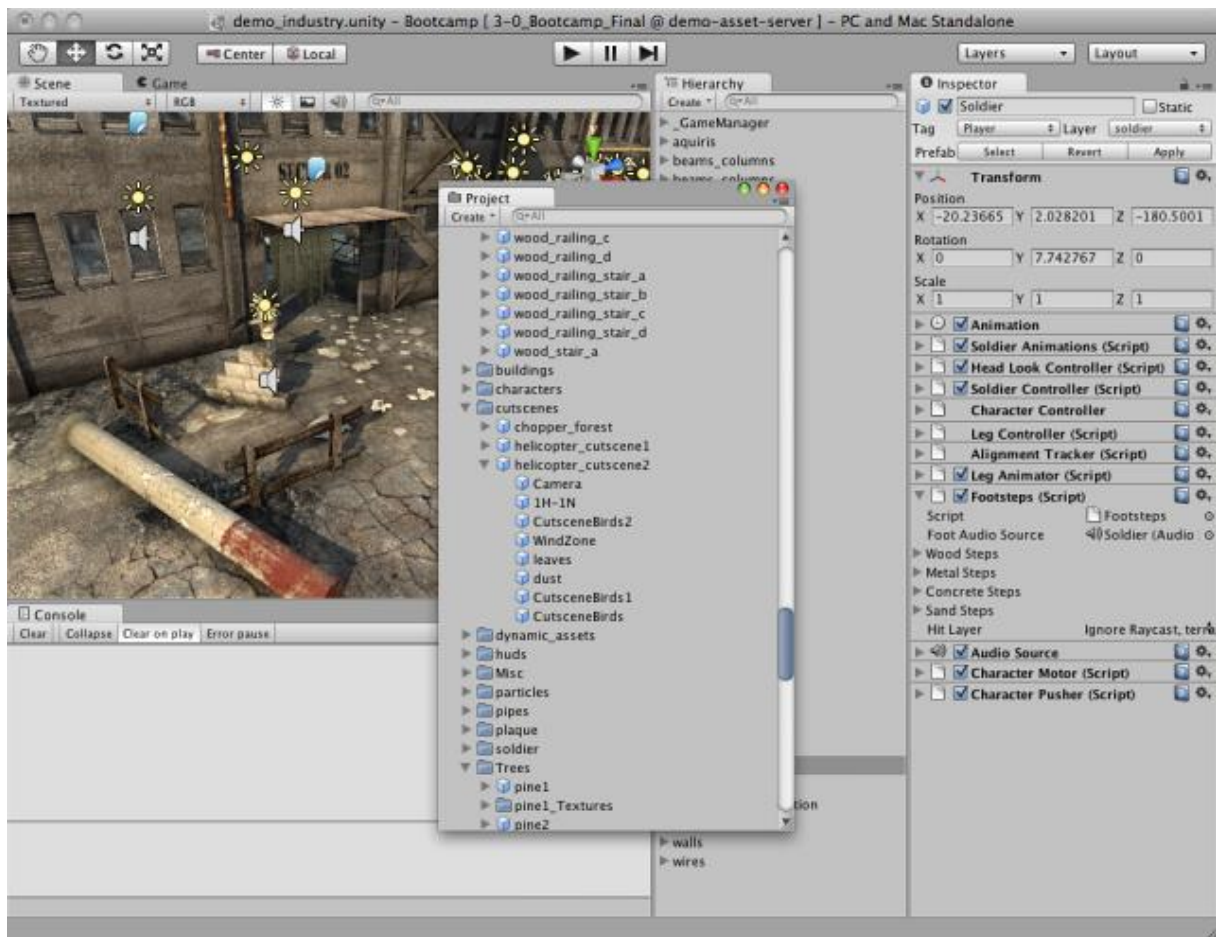


Рис. 2.1. Робоче вікно ігрового рушія Unity3D

CryEngine [12]– найпотужніший, але вузьконаправлений на проекти типу «Шутер» ігровий рушій. За його допомогою можна забезпечити у проекті фотореалістичну графіку. Підтримує графічну бібліотеку DirectX 12 та програмування шейдерів [13], які необхідні для роботи з параметрами об'єктів, і зображеннями тривимірної графіки. Для розроблення дозволяє використовувати мову програмування C++ [14].

C++ є основною мовою програмування для розроблення ігор, бо за допомогою цієї мови програмування можна забезпечити повний контроль над усіма ресурсами, такими як, наприклад, використовувана пам'ять, у проекті. Має багато існуючих бібліотек та є лідером по швидкості та кросплатформності. Але, незважаючи на велику кількість існуючих бібліотек, є достатньо важкою для засвоєння через необхідність слідкувати

за всіма ресурсами самостійно для забезпечення надійності роботи програми.

Серед недоліків цього ігрового рушія слід зазначити, що він потребує найбільшу кількість ресурсів і завдяки основному фокусу на реалізмі, створити легковісну програму з допомогою цього рушія майже неможливо. Інтерфейс робочого вікна рушія наведений на рис. 2.2

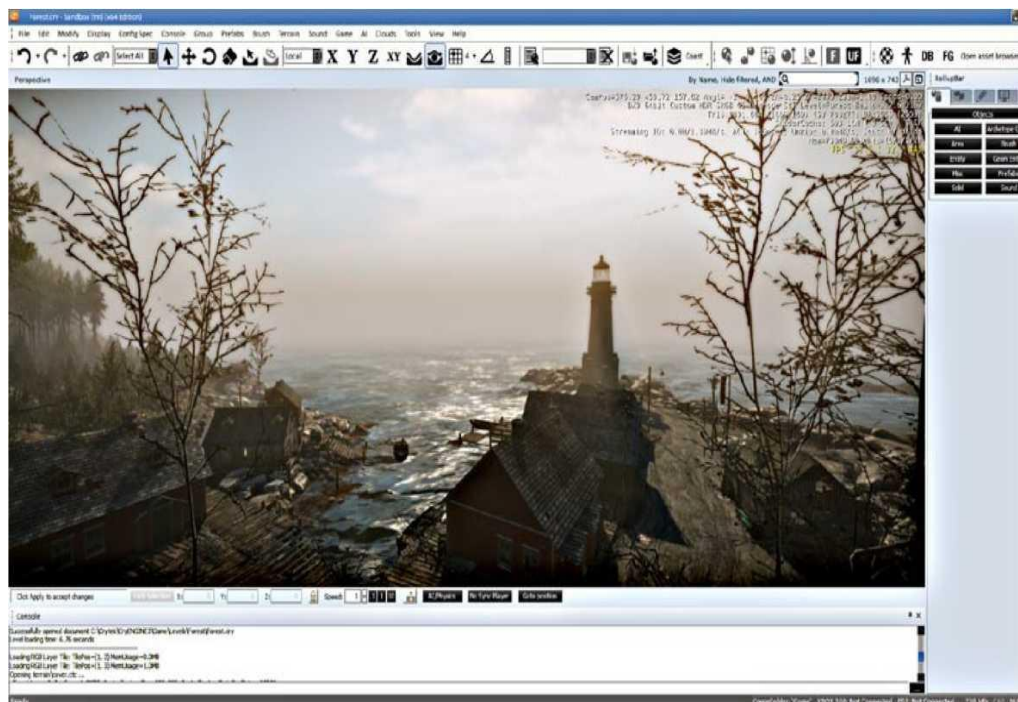


Рис. 2.2. Робоче вікно ігрового рушія CryEngine

Unreal Engine [15] – ігровий рушій з великим різноманіттям інструментарію для 2D та 3D проєктів. Мовою програмування є C++, який був описаний вище. Так само, як і Unity3D, є кросплатформним, підтримує різні графічні бібліотеки, такі як DirectX та OpenGL та має постійну підтримку у вигляді великих і регулярних оновлень. Завдяки великій кількості інструментарію, можна досягти високої якості графіки, але проєкт буде не таким легковісним, як за допомогою Unity3D. З тих самих причин має більш складний для початківців інтерфейс (див. рис. 2.3).

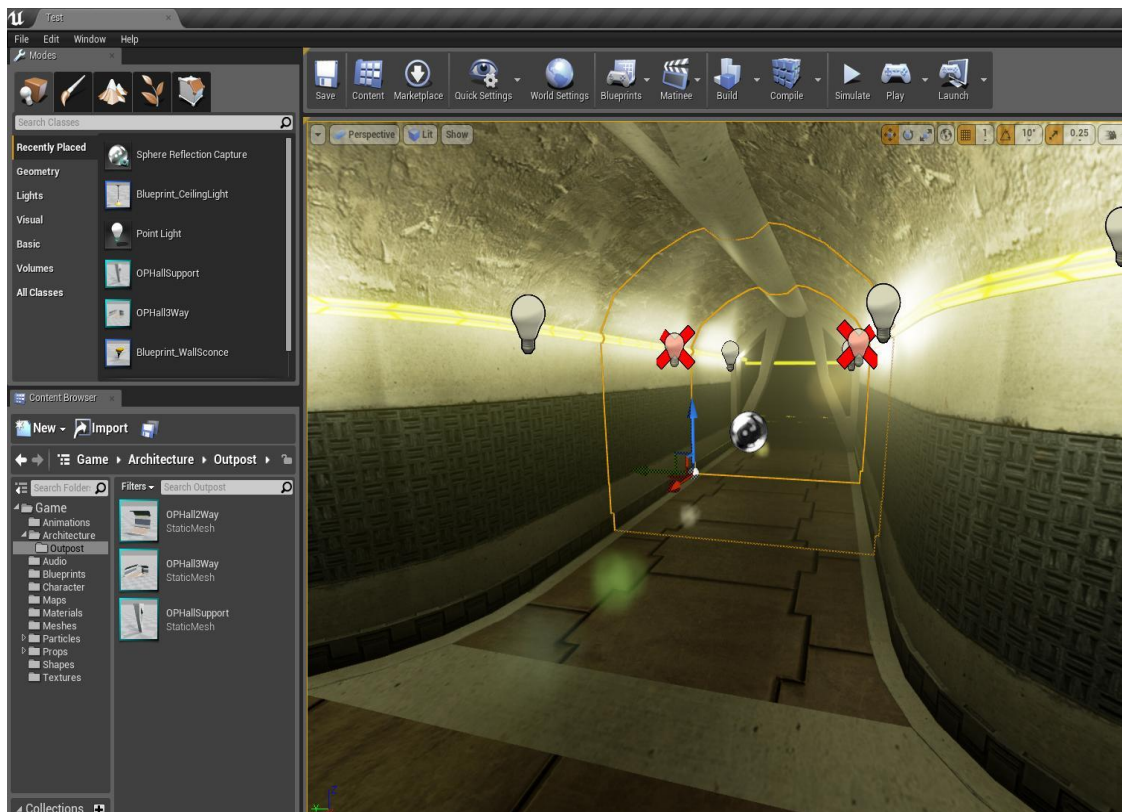


Рис. 2.3. Робоче вікно ігрового рушія Unreal Engine

Cocos2D-x [16] – ігровий рушій з відкритим вихідним кодом. Не має вбудованого редактора ігрових сцен, що робить розроблення проекту складнішою. Розробникам надається вибір з двох мов програмування: C++ та JavaScript. Є кросплатформним, але перелік підтримуваних платформ значно менший, ніж у більш відомих ігрових рушіїв.

Проведемо аналіз таких характеристик ігрових рушіїв:

- кросплатформність;
- легковісність;
- простота використання.

Таблиця 2.1

Порівняння ігрових рушіїв за кросплатформністю та легковісністю

| Ігровий рушій | Кросплатформність | Легковісність (за пріоритетами) |
|---------------|---------------------------------|------------------------------------|
| Unity3D | Повна підтримка | 2 |
| CryEngine | Повна підтримка | 4 |
| Unreal Engine | Повна підтримка | 3 |
| Cocos2D-x | iOS, Android, WoPhone, Win32 | 1 |

Таблиця 2.2

Порівняння ігрових рушіїв за простотою використання

| Ігровий рушій | Простота використання (за пріоритетами) |
|---------------|--|
| Unity3D | 1 |
| CryEngine | 3 |
| Unreal Engine | 2 |
| Cocos2D-x | 4 |

Отже, можна зробити висновок, що найкращим ігровим рушієм для розроблюваного програмного забезпечення буде Unity3D, оскільки він є найпростішим у використанні та достатньо легковісним.

2.2. Вибір API для використання технології розпізнавання мови

Аналіз існуючих технологій розпізнавання мови проведено лише серед продуктів найвідоміших компаній, таких як Amazon, Google та Microsoft, оскільки при розробленні додатку для дітей необхідна висока точність результату розпізнавання, а у відомих компаній багато даних для

тренування своїх нейронних мереж. Характеристики, за якими проводився аналіз, обрані такі:

- Кросплатформність. Оскільки розроблюване програмне забезпечення повинно бути кросплатформним, ця вимога поширюється і на технологію розпізнавання мови.
- Підтримка російської та української мови. Оскільки цільова аудиторія програмного забезпечення – діти молодших класів українських загальноосвітніх шкіл, даний критерій є обов’язковим, бо більшість технологій розпізнавання мови підтримують лише англійську мову.
- Тривалість безкоштовного випробувального періоду. Для впевненості у точності вихідних даних та проведення аналізу правильності роботи програми вкрай бажаним є безкоштовний випробуваний термін перед оплатою.
- Підтримувані аудіо формати.

Розглянемо нижче існуючі технології відповідно до вказаних вище вимог.

- Amazon Transcribe [17]
 - може працювати на будь-яких пристроях з мікрофоном;
 - нема підтримки ні російської, ні української мови;
 - випробуваний термін присутній, 60 хвилин на місяць впродовж 12 місяців;
 - підтримувані аудіо формати: .MP3 та .WAV.
- Google Cloud Speech-to-Text [18]
 - кросплатформений;
 - підтримує російську та українську мову;
 - присутній випробуваний термін тривалістю в рік або по закінченню ліміту в 300 доларів;
 - підтримує аудіо файли з розширенням .WAV.

- Microsoft Azure Speech to Text [19]

- не є кросплатформеним, має підтримку лише пристроїв з встановленим Speech SDK [20];
- має підтримку лише російської мови;
- 30-денний випробуваний період;
- підтримує аудіо файли з розширенням .WAV.

Отже, беззаперечним лідером серед розглянутих технологій розпізнавання мови є Google Cloud Speech-to-Text, який і буде використаний в розроблюваному програмному забезпеченні.

3. СТРУКТУРНО-АЛГОРИТМІЧНА ОРГАНІЗАЦІЯ

3.1. Аналіз вимог до програмних засобів

Розроблюване програмне забезпечення складається з сукупності ігрових сцен, які пов'язані між собою сценою головного меню.

Для наочності систематизуємо всі вимоги, висунуті до програмного забезпечення для підтримки дидактичної діяльності викладачів молодших класів загальноосвітньої школи за допомогою стандарту *IDEF0* [21].

Як було визначено раніше, мета даного продукту – забезпечити підтримку всіх досліджених категорій процесів розвитку дитини, включаючи соціально-психологічну.

Визначимо вхідні та вихідні дані. До вхідних даних належить інформація, отримана в результаті досліджень та аналізу існуючих рішень, а саме визначення категорій процесів розвитку дитини та приклади існуючих продуктів, впливаючих на гностичні та інструментальні процеси розвитку дитини, на які ми будемо спиратись в якості прикладу. В результаті ми маємо отримати програмне забезпечення, націлене на тренування всіх категорій процесів розвитку учня молодших класів загальноосвітньої школи.

Виділимо необхідні для розроблення даного ігрового комплексу механізми:

- розробник;
- ігровий рушій Unity 3D;
- доступ до API технології розпізнавання голосу, а саме Google Cloud Speech-to-Text;
- ігрові ассети [22] для розроблення приємного дитячому оку інтерфейсу;
- доступ до Інтернету для коректної роботи технології розпізнавання голосу.

Вимогами до розроблення даного проекту є:

- вимоги, сформовані UI та UX дизайнерами;
 - якщо гра впливає на інструментальні процеси розвитку дитини – кількість елементів на екрані має бути мінімальною;
 - якщо гра впливає на гностичні процеси розвитку дитини – основною вимогою до інтерфейсу є привабливість;
 - якщо гра впливає на соціально-психологічні процеси розвитку дитини – інтерфейс має бути привабливим та не перевантаженим.
- вимоги до організації навчального процесу у ігровій формі, сформовані авторами досліджених педагогічних статей [1];
 - створення ігрової атмосфери;
 - проведення гри, в результаті якої повинна бути виконана поставлена задача;
 - підведення підсумків.
- вимоги до структури організації сцен у ігровому рушії Unity 3D.

Діаграма IDEF0, сформована за допомогою всіх описаних вище складових, представлена на рис. 3.1.

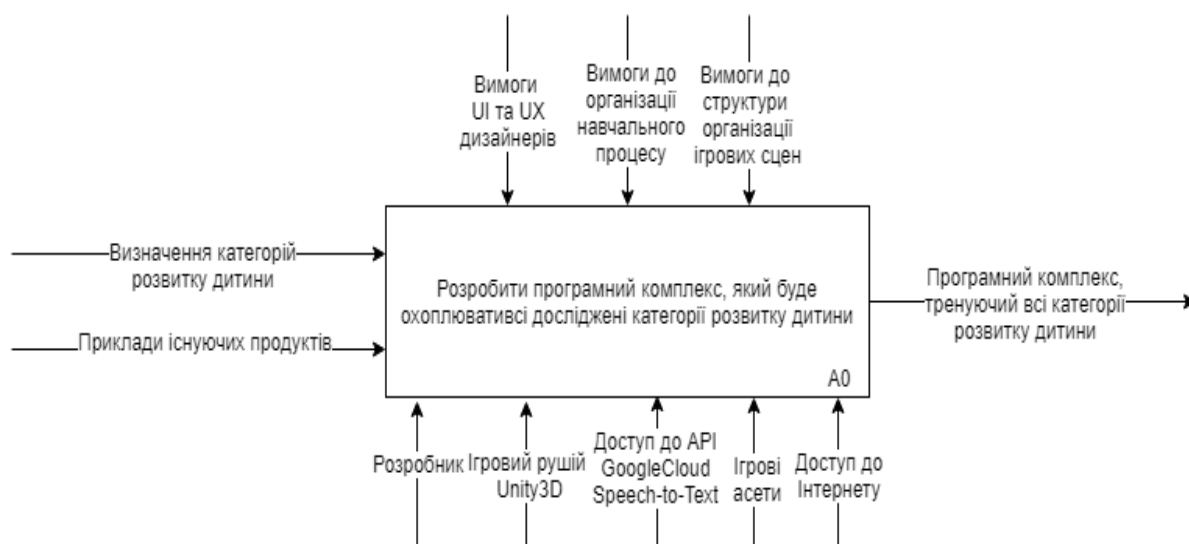


Рис. 3.1. Діаграма IDEF0

3.1.1. Вимоги до соціально-психологічної складової програмного забезпечення

Складовою, що забезпечує підтримку соціально-психологічної функції програмного забезпечення, є гра на вивчення школярами правил базової ввічливості в ігровій формі за допомогою технології розпізнавання мови Google Cloud Speech-to-Text. Функціональні та нефункціональні вимоги у вигляді реєстру представлені в табл. 3.1.

Таблиця 3.1

Реєстр вимог до соціально-психологічної складової програмного додатку

| Код вимоги | Зміст вимоги | Атрибути | | |
|------------|--|-----------|------------|---------------|
| | | Пріоритет | Складність | Тип |
| 001 | Збереження запису з мікрофону до тимчасового файлу з розширенням .WAV | Високий | Висока | Функціональна |
| 002 | Парсинг відповіді, отриманої з API | Високий | Висока | Функціональна |
| 003 | Забезпечення стійкості системи шляхом перевірки на коректність отриманих даних | Високий | Висока | Функціональна |
| 004 | Вибір найбільш точного результату з наданих в результаті запиту до API | Високий | Середня | Функціональна |
| 005 | Після оброблення результату тимчасовий файл має бути видалений | Високий | Середня | Функціональна |
| 006 | Співставлення отриманої відповіді з очікуваною | Високий | Середня | Функціональна |

| Код вимоги | Зміст вимоги | Атрибути | | |
|------------|--|-----------|------------|-----------------|
| | | Пріоритет | Складність | Тип |
| 007 | Повторення поточного завдання у випадку помилки та вибір наступного у випадку правильної відповіді | Високий | Середня | Функціональна |
| 008 | Створення мінімалістичного інтерфейсу | Середня | Висока | Нефункціональна |

3.1.2. Вимоги до інструментальної складової програмного забезпечення

Складовою, що забезпечує підтримку інструментальної функції програмного забезпечення, є гра на навчання швидкому друку. Функціональні та нефункціональні вимоги у вигляді реєстру представлені в табл. 3.2.

Таблиця 3.2

Реєстр вимог до інструментальної складової програмного забезпечення

| Код вимоги | Зміст вимоги | Атрибути | | |
|------------|--|-----------|------------|---------------|
| | | Пріоритет | Складність | Тип |
| 001 | Повільна поява на екрані нових слів, які поступово зсуваються вниз | Високий | Висока | Функціональна |

| Код вимоги | Зміст вимоги | Атрибути | | |
|------------|--|-----------|------------|-----------------|
| | | Пріоритет | Складність | Тип |
| 002 | Підсвічування поточного слова та зникнення набраних букв з нього | Високий | Висока | Функціональна |
| 003 | Поява меню перезавантаження гри у випадку певної кількості пропущених слів | Високий | Висока | Функціональна |
| 004 | Вибір приємного для ока шрифту для падаючого тексту | Високий | Середня | Нефункціональна |

3.1.3. Вимоги до гностичної складової програмного забезпечення

Складовою, що забезпечує підтримку гностичної функції програмного забезпечення, є гра на пошук предметів. Функціональні та нефункціональні вимоги у вигляді реєстру представлені в табл. 3.3.

Таблиця 3.3

Реєстр вимог до гностичної складової програмного забезпечення

| Код вимоги | Зміст вимоги | Атрибути | | |
|------------|---|-----------|------------|-----------------|
| | | Пріоритет | Складність | Тип |
| 001 | Завантаження ігрових об'єктів для пошуку | Високий | Висока | Функціональна |
| 002 | Створення панелі з силуетами об'єктів, які необхідно знайти | Високий | Висока | Функціональна |
| 003 | Автоматичне видалення знайдених об'єктів з нижньої панелі | Високий | Середня | Функціональна |
| 004 | Створення інтерфейсу, на якому буде складно помітити потрібні об'єкти | Високий | Висока | Нефункціональна |

3.2. Модуль підтримки соціально-психологічних процесів розвитку дитини

Програмний модуль, що впливає на соціально-психологічні процеси розвитку дитини, розроблений у вигляді окремої ігрової сцени і є єдиним модулем, який потребує підключення до Інтернет. Використана технологія розпізнавання мови від компанії Google містить у своїй документації опис свого API, за допомогою якого при завантаженні на хмарний сервіс записаного аудіо запису відбувається його аналіз за допомогою технологій машинного навчання, який автоматично визначає мову з підтримуваних, що складає близько 120 варіантів. Логіка модуля, згідно до вимог створення сцен в ігровому рушії Unity 3D, розроблена на мові програмування C#.

3.2.1. Алгоритм роботи модулю підтримки соціально-психологічних процесів розвитку дитини

Крок 1. Відбувається випадковий вибір вправи для виконання з доступних у модулі.

Крок 2. Програється відповідний йому аудіо запис у вигляді вірша, де школяру треба записати відповідне слово ввічливості, яке є завершенням вірша.

Крок 3. Очікується натискання на кнопку для початку запису відповіді школяра та повторне натиснення для збереження цього файлу у тимчасовий аудіо файл з розширенням .WAV.

Крок 4. Відбувається завантаження тимчасового аудіо запису на хмарний сервіс для аналізу.

Крок 5. Проводиться парсинг отриманої з хмарного сервісу відповіді.

Крок 6. Проводиться пошук найбільш точної відповіді з отриманих варіантів.

Крок 7. Аналізується правильність відповіді школяра.

Крок 8. Відбувається автоматичний запуск наступної вправи у випадку правильної відповіді, а в іншому випадку повторення вправи.

Блок-схема алгоритму роботи модуля зображена на рис. 3.3.

Розглянемо методи, які використовуються у наведеній нижче блок-схемі:

- `MicrophoneConnected()` – перевіряє на наявність підключеного мікрофону;
- `RecordingStarted()` – перевіряє на стан мікрофону, чи ведеться запис у поточний момент часу;
- `RecordingFinished()` – перевіряє на стан кнопки, натискання на яку буде сигналізувати про намір завершити запис;
- `SaveTempFile()` – зберігає запис у тимчасовий аудіо файл;
- `UploadFileToCloud()` – виконує запит до Google Cloud Speech-To-Text;

- `ParseResult()` – парсить відповідь від сервісу розпізнавання мови та обирає ту, яку сервіс позначив як найбільш вірогідну;
- `RestartLesson()` – повторює ту саму вправу;
- `StartNewLesson()` – випадковим способом обирає іншу вправу з можливих та запускає її.
- `Show Label(string)` – виводить вказаний в якості параметру рядок на екран;
- `PlayAdioLesson()` – запускає аудіо запис з вправою;
- `BackButtonPressed()` – перевіряє стан кнопки, яка забезпечує повернення до головного меню.

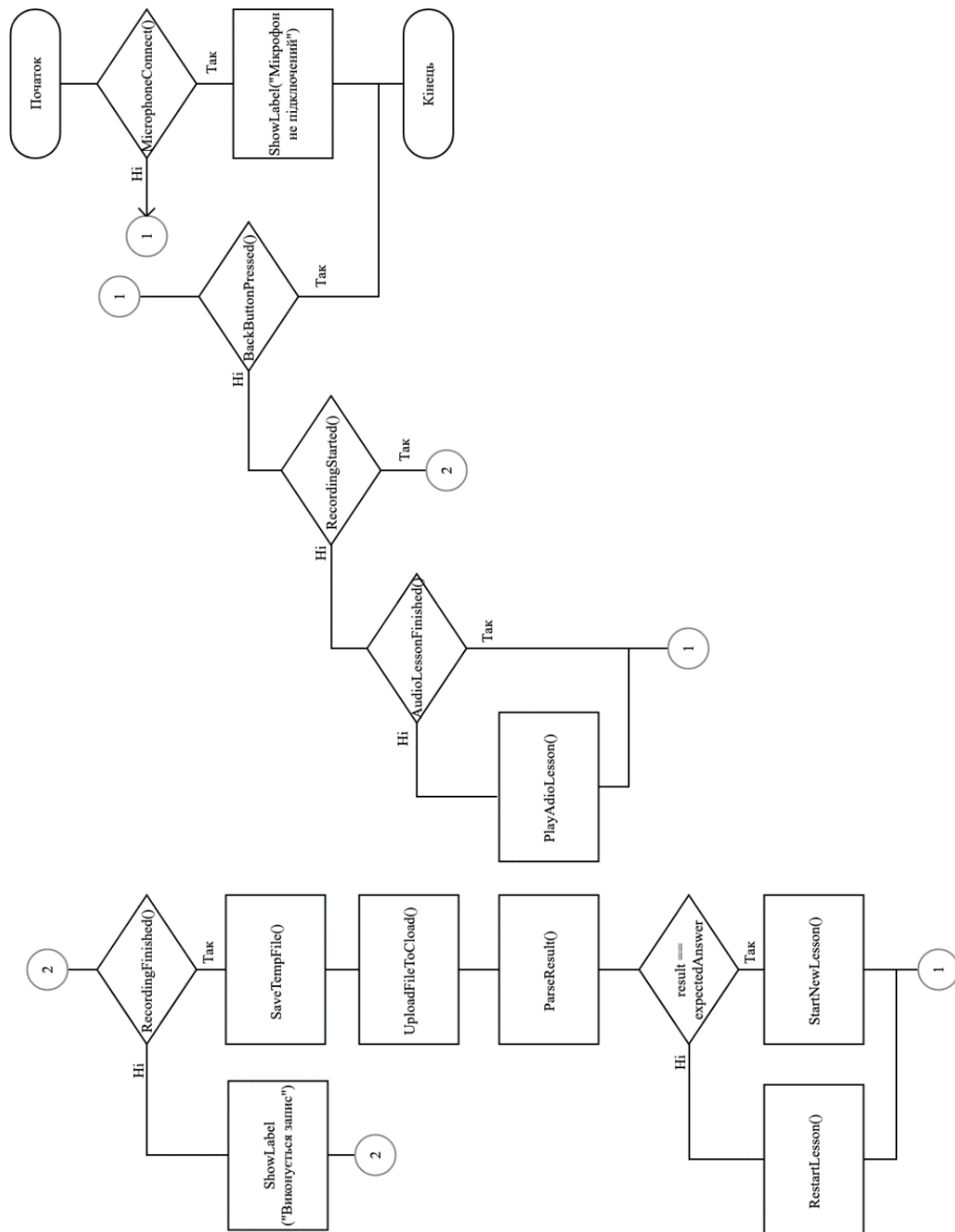


Рис. 3.3. Блок-схема алгоритму роботи модулю підтримки соціально-психологічних процесів розвитку дитини

3.3. Модуль підтримки гностичних процесів розвитку дитини

Програмний модуль, впливаючий на гностичні процеси розвитку дитини, також розроблений у вигляді окремої ігрової сцени і є модулем з найбільш деталізованим користувацьким інтерфейсом. Містить в собі декілька рівнів, тобто декілька сцен, які завантажуються у динамічний спосіб, щоб зменшити час побудови сцен. Логіка модуля, згідно до вимог створення сцен в ігровому рушії Unity 3D, розроблена на мові програмування C#.

3.3.1. Алгоритм роботи модулю підтримки гностичних процесів розвитку дитини

Крок 1. З'являється сцена головного меню, де користувач може обрати рівень гри перед початком та дві кнопки: вихід та початок гри.

Крок 2. На початку гри завантажується сцена, інтерактивні об'єкти та силуети цих об'єктів для відображення у нижній панелі, як підказки, які предмети необхідно шукати.

Крок 3. Якщо список об'єктів для пошуку не пустий, чекаємо, доки буде натиснутий один з шуканих предметів.

Крок 4. При коректному натисканні на об'єкт – видаляємо його зі сцени та панелі з силуетами.

Крок 5. Коли всі потрібні об'єкти знайдені – з'являється меню закінчення гри, де можливо почати гру з початку або вийти з гри.

Блок-схема алгоритму роботи модуля зображена на рис. 3.4.

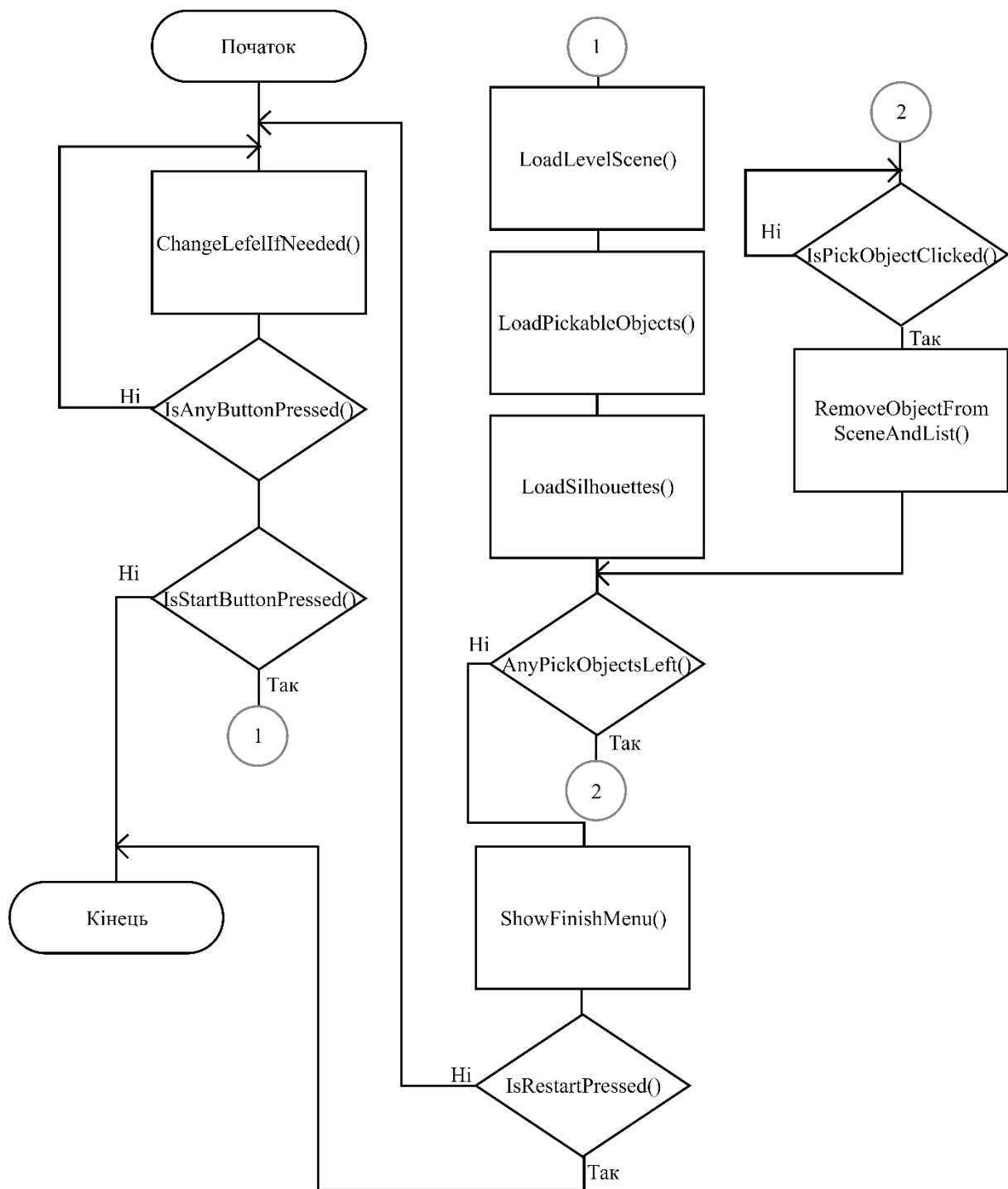


Рис. 3.4. Блок-схема алгоритму роботи модулю підтримки гностичних процесів розвитку дитини

Розглянемо методи, наведені у блок-схемі:

- `ChangeLevelIfNeeded()` – змінює рівень гри на вибраний з випадального списку;

- Exit() – вихід з гри, повернення до головного меню програмного забезпечення;
- Start() – початок нової гри;
- LoadLevelScene() – завантажує обрану в головному меню гри сцену;
- LoadSilhouettes() – генерує список об'єктів, що будуть доступні для вибору та завантажує їхні силуети на нижню панель;
- IsAnyAvailableToPickObjectsLeft() – перевіряє, чи є ще об'єкти, які користувачу необхідно знайти;
- IsAvailableToPickObjectClicked() – перевіряє, чи силует натиснутого об'єкту є у нижній панелі силуетів;
- RemoveObjectFromSceneAndList() – видаляє об'єкт та його силует зі сцени;
- ShowFinishMenu() – завантажує сцену успішного закінчення гри. Містить дві кнопки: початок гри з початку та вихід з гри до головного меню програмного забезпечення;
- IsRestartButtonPressed() – перевіряє, чи натиснутою кнопкою є кнопка перезапуску гри з початку;

3.4. Модуль підтримки інструментальних процесів розвитку дитини

Програмний модуль, впливаючий на інструментальні процеси розвитку дитини, також розроблений у вигляді окремої ігрової сцени і є модулем, який розроблений з найбільш простим користувацьким інтерфейсом, щоб зосередити увагу дитини лише на завданні, а саме на повільно падаючих вниз словах, які необхідно набирати на клавіатурі. Логіка модуля, згідно до вимог створення сцен в ігровому рушії Unity 3D, розроблена на мові програмування C#.

3.4.1. Алгоритм модулю підтримки інструментальних процесів розвитку дитини

Крок 1. Кожен константний проміжок часу з масиву зарані підготовлених слів випадковим способом обирається слово.

Крок 2. Вибране слово відображається на екрані у випадково обраному місці по осі X в межах екрану і кожен кадр його позиція зміщується вниз по осі Y.

Крок 3. При натисканні на клавішу на клавіатурі, що відповідає поточній букві слова:

- слово підсвічується іншим кольором як поточне слово;
- поточна буква зникає зі слова;
- при досягненні певної кількості пропущених слів, з'являється екран закінчення з наступними кнопками:
- початок нової гри;
- повернення до головного меню.

Блок-схема алгоритму роботи модуля зображена на рис. 3.5.

Розглянемо методи, які використовуються у наведеній нижче блок-схемі:

- `SpawnWord()` – метод, який додає на екран випадково обране з зарані підготовленого масиву слів наступне;
- `IsLimitReached()` – метод, який визначає, чи досягнутий ліміт пропущених слів;
- `IsActiveWordAvailable()` – метод, який перевіряє, чи є вже слово, яке почали друкувати;
- `IsFirstLetterTyped()` – метод, який перевіряє, чи правильно натиснута чергова клавіша слова на клавіатурі;
- `RemoveLetter()` – видаляє поточну(першу) літеру зі слова;
- `IsAnyLettersAvailable()` – перевіряє, чи є ще літери в поточному слові;
- `RemoveWord()` – видаляє об'єкт-слово з поточного масиву об'єктів на екрані;
- `ShowPauseMenu()` – виконує видалення всіх об'єктів-слів на екрані та відображає дві кнопки навігації: повернення до головного меню та повторна спроба;

- HighlightActiveWord() – виділяє активне слово червоним кольором;
- IsRestartButtonClicked() – визначає, яка з кнопок при завершенні гри була натиснута.

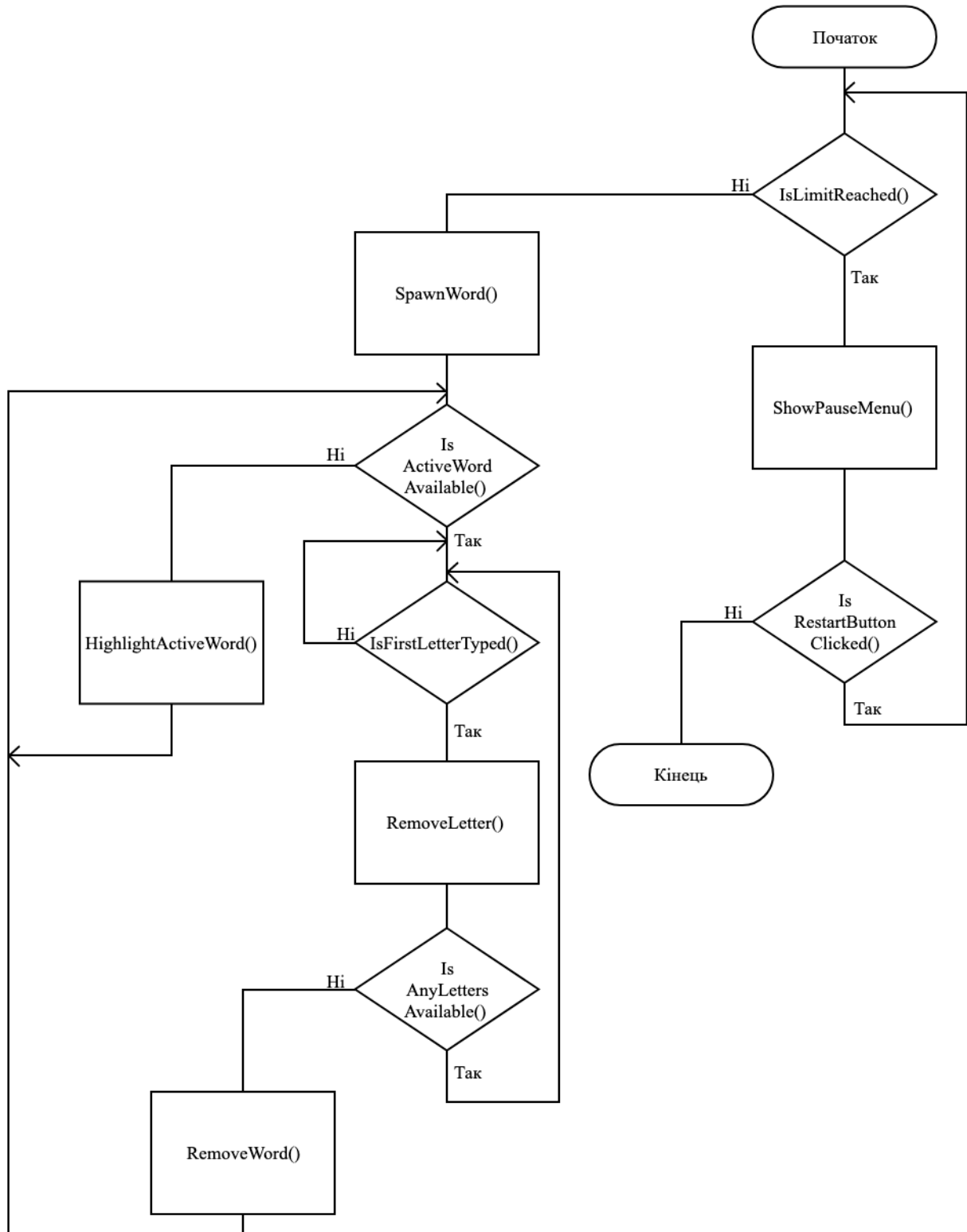


Рис. 3.5. Блок-схема алгоритму роботи модулю підтримки інструментальних процесів розвитку дитини

4. АНАЛІЗ РОЗРОБЛЕНИХ ПРОГРАМНИХ ЗАСОБІВ

4.1. Особливості реалізації

4.1.1. Особливості створення програмного забезпечення за допомогою ігрового рушія Unity 3D

Даний ігровий рушій передбачає застосування компонентно-орієнтованого підходу [23], що означає, на відміну від об'єктно-орієнтованого підходу, що об'єкт буде описаний не класом, а набором властивостей в його компоненті. Даний підхід дозволяє значно зменшити кількість дуплікації коду, оскільки один і той самий компонент можна використовувати у багатьох об'єктах. За допомогою компонентно-орієнтованого підходу розширення системи в майбутньому стає достатньо «дешевим» з точки зору людино-годин, оскільки для внесення нового об'єкту в ієрархію не потрібно вносити зміни до існуючої архітектури, що може бути необхідним у об'єктно-орієнтованому підході.

4.1.2. Програмний модуль, що впливає на гностичні процеси розвитку дитини

Програмний модуль, що впливає на гностичні процеси розвитку дитини, ґрунтується на роботі з голосом, тому потребує наявності мікрофону в пристрої, з якого запускається дане програмне забезпечення. Для виведення аудіо запису за допомогою звукового каналу у ігровому рушії Unity використовується компонент під назвою AudioSource [24], а для запису даних з мікрофону використовується стандартний для ігрового рушія клас Microphone [25]. До кожного інтерактивного об'єкту, у випадку соціально-психологічного модуля це лише кнопки для запису відповіді та повернення до головного меню, додаються додаткові компоненти, а саме написані на мові програмування C# скрипти, в яких описана логіка кожного етапу роботи з даними. Також до ігрової сцени належать декілька текстових полів, в які відбувається запис тексту поточної вправи, окрім виведення

цього ж тексту через аудіо канал; відбувається візуалізація у текстовому вигляді отриманої, обробленої та проаналізованої відповіді від хмарного сервісу розпізнавання мови Google Speech-To-Text на надісланий аудіо файл з відповіддю на поточну вправу.

4.1.3. Програмний модуль, що впливає на інструментальні процеси розвитку дитини

Програмний модуль, що впливає на інструментальні процеси розвитку дитини, ґрунтується на обробці даних, введених з клавіатури, тому ніяких особливих вимог не потребує. На сцену кожен константний проміжок часу додаються нові об'єкти класу WordDisplay, який містить у собі стандартне текстове поле, до якого додається текст, що є випадково обраним словом з можливих, зарані визначених, та з якого видаляється перша літера кожен раз, коли вона натиснута на клавіатурі. При початку набору слова до тексту застосовується стилістичний ефект, щоб виділити його з-поміж інших як активне слово. Сама сцена містить скрипт з назвою WordManager, який контролює поточний стан всіх слів та містить масив всіх пропущених користувачем слів. Як тільки кількість пропущених слів перевищує заздалегідь визначене константне значення – ігровий процес завершується, всі слова з екрану зникають та з'являється меню, яке дозволяє виконати наступні дії: спробувати ще раз або повернутися до головного меню ігрового комплексу. До кожного з цих об'єктів також доданий скриптовий компонент, в якому описана його логіка поведінки.

4.1.4. Програмний модуль, що впливає на гностичні процеси розвитку дитини

Даний модуль ґрунтується на обробці даних, введених з комп'ютерної миші, тому ніяких особливих вимог не потребує. На початку гри користувачу надається можливість обрати рівень гри та почати її, або повернутись назад, до головного меню. На початку ігрового рівня динамічним образом завантажується сцена. Під динамічним способом мається на увазі, що завантажується не створена у ігровому редакторі сцена,

з зарані підготовленими ігровими спрайтами, а проводиться читання текстового файлу, який містить інформацію про розмір ігрової сцени та характеристики всіх її компонентів. Таким чином, зменшується тривалість завантаження ігрової сцени. В даному випадку це необхідно через велику кількість об'єктів, які мають бути присутніми на ігровій сцені. Для забезпечення такого способу завантаження сцени необхідним є створення так званих префабів, які не є реальними ігровими об'єктами, а лише повним їх інтерфейсом. До кожного з цих префабів в якості компонентів додано скрипти, які визначають логіку поведінки. Найбільш важливим це є для інтерактивних об'єктів, які можуть бути вибрані у випадку наявності їхніх силуетів на нижній панелі підказок.

4.2. Інтерфейс та вміст модулів

Згідно з поставленими у першому розділі вимогами до розвиваючих програмних засобів, дизайн кожного типу ігор має відрізнятись один від одного для досягнення мети кожного з типів модулів. Від головного меню вимагалась лише привабливість інтерфейсу (рис. 4.1).



Рис. 4.1. Інтерфейс головного меню

4.2.1. Модуль, що впливає на соціально-психологічні процеси розвитку дитини.

Відповідно до вимог, інтерфейс даного типу модулю має бути привабливим, але не перевантаженим великою кількістю елементів. Тому на ігровій сцені присутня мінімальна кількість елементів, а саме:

- текст вправи;
- невелике зображення, яке асоціюється із вправою;
- кнопка для запису відповіді;
- кнопка для повернення до головного меню.

Зображення інтерфейсу модуля зображене на рис. 4.2.



Рис. 4.2. Інтерфейс модулю, що впливає на соціально-психологічні процеси розвитку дитини

4.2.2. Модуль, що впливає на інструментальні процеси розвитку дитини

Згідно з вимогами інтерфейс даного модулю має бути сфокусований лише на вправі, оскільки за допомогою даного додатку мають формуватися

основні навички та вміння, саме цьому більшу частину екрану в розробленому програмному модулі займають слова.

Зображення інтерфейсу модуля зображене на рис. 4.3.

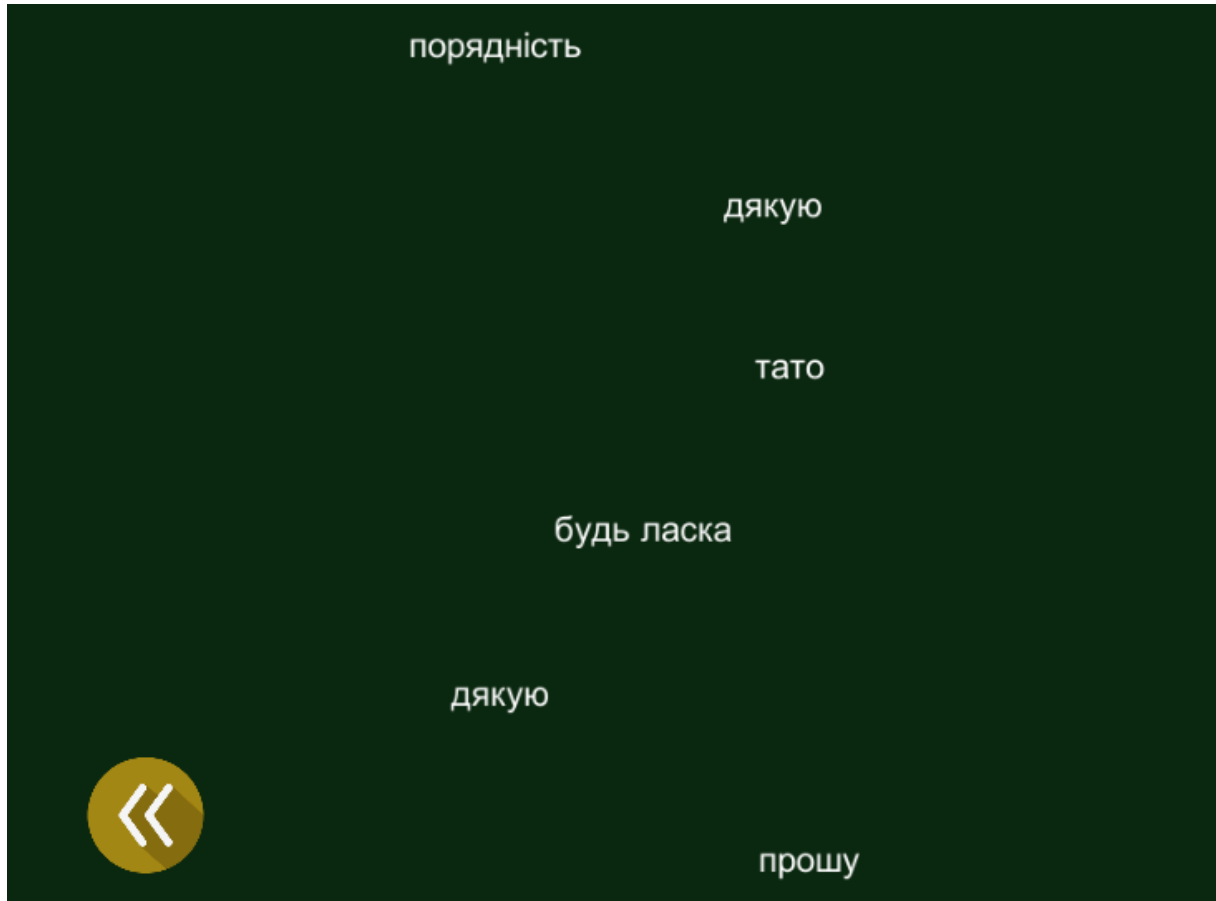


Рис. 4.3. Інтерфейс модулю, що впливає на інструментальні процеси розвитку дитини

4.2.3. Модуль, що впливає на гностичні процеси розвитку дитини

Даний модуль повинен мати найбільш привабливий інтерфейс, оскільки в даній грі планується, що увага дитини має бути сфокусована на зображеннях.

Даний модуль включає в себе декілька сцен, а саме сцену головного меню, кожен з двох ігрових сцен та сцену успішного завершення гри.

Розглянемо кожен ігрову сцену на рис. 4.4. – 4.7.

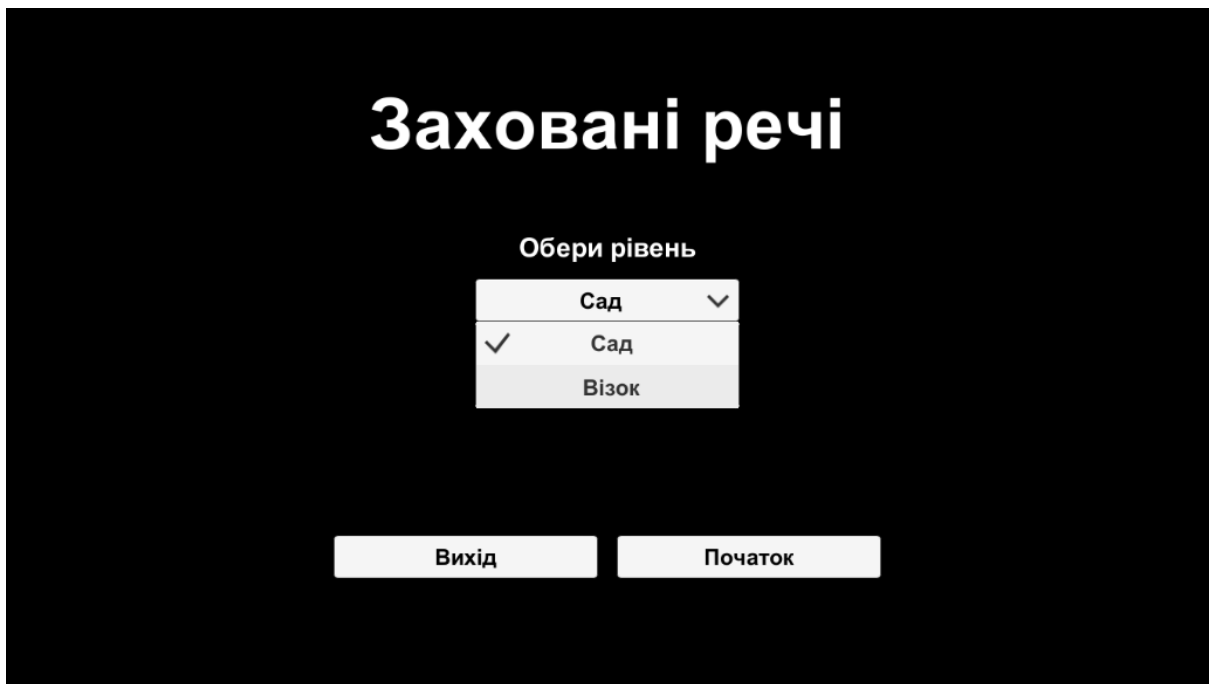


Рис. 4.4. Інтерфейс головного меню

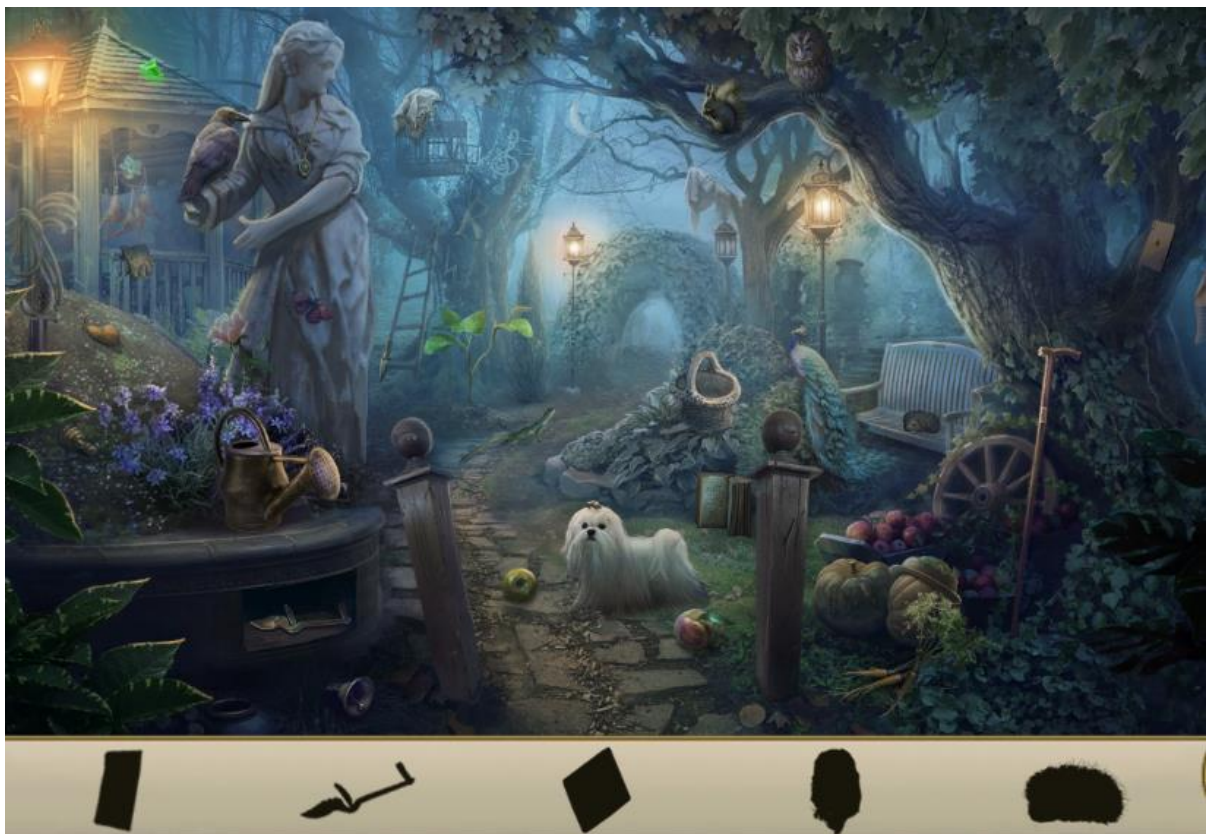


Рис. 4.5. Інтерфейс ігрової сцени «Сад»



Рис. 4.6. Інтерфейс ігрової сцени «Візок»



Рис. 4.7. Інтерфейс екрану успішного завершення гри

4.3. Тестування програмного забезпечення

Для впевненості, що програмне забезпечення задовольняє всім поставленим вимогам та функціонує правильно, було розроблено набір тест-кейсів димового тестування.

Сценарій димового тестування:

1. Запустити програмне забезпечення.
2. Почати роботу з модулем, що впливає на соціально-психологічні процеси розвитку дитини, закінчити одну вправу. Повернутись до головного меню.
3. Почати роботу з модулем, що впливає на інструментальні процеси розвитку дитини. Перевірити коректність відображення вводу слів. Дочекались прогашу, повернутись до головного меню.
4. Почати роботу з модулем, що впливає на гностичні процеси розвитку дитини. Завершити гру в кожному з режимів. Повернутись до головного меню.
5. Завершити роботу програмного забезпечення.

Розглянемо тест-кейси в табл. 4.1.

Таблиця 4.1

Тест-кейси до розробленого програмного забезпечення

| № п/п | Опис | Очікувані результати |
|-------|---|---|
| 1 | Запустити програмне забезпечення, два рази натиснувши на відповідний ярлик. | Програмне забезпечення запускається без помилок, відображається головне меню. |

| № п/п | Опис | Очікувані результати |
|----------|---|---|
| 2 | <ol style="list-style-type: none"> 1. Натиснути на кнопку «Соціально-психологічна гра» 2. Натиснути на кнопку «Записати відповідь» 3. Натиснути на кнопку «перевірити» 4. Натиснути на кнопку «Назад» | <ol style="list-style-type: none"> 1. Відкривається ігрова сцена. Запускається аудіо запис із вправою, текст із аудіо запису відображається на екрані, з'являється зображення, яке асоціюється із вправою 2. Текст на кнопці «записати відповідь» змінюється на «перевірити». 3. У випадку правильної відповіді програвся аудіо запис іншої вправи, змінюється текст і зображення, у верхньому лівому кутку екрана з'являється надпис користувацької відповіді з позначкою «правильно». В іншому випадку відбувається повторення вправи, у верхньому лівому кутку екрана з'являється надпис користувацької відповіді з позначкою «неправильно». 4. З'являється головне меню розвиваючого програмного забезпечення |
| 3 | <ol style="list-style-type: none"> 1. Натиснути на кнопку «Інструментальна гра» 2. Почати набирати «падаюче» слово 3. Закінчити набирати «падаюче» слово 4. Почекаати, доки декілька слів спустяться нижче області екрана 5. Натиснути на кнопку «назад» під час гри або в меню завершення гри | <ol style="list-style-type: none"> 1. Відкривається ігрова сцена. З верхньої частини екрану починають повільно «падати слова» 2. Те слово, що користувач почав набирати, становиться іншого кольору, зникають набрані букви 3. Зникає все слово 4. З екрану зникають всі слова, з'являється екран паузи. 5. Виконується перехід до головного меню розвиваючого програмного забезпечення. |

| № п/п | Опис | Очікувані результати |
|-------|--|---|
| 4 | <ol style="list-style-type: none"> 1. Натиснути на кнопку «гностична гра» 2. Обрати тип сцени «Сад» 3. Знайти перший предмет 4. Знайти всі предмети 5. Натиснути на кнопку «Почати спочатку» 6. Обрати тип сцени «візок» 7. Знайти перший предмет 8. Знайти останній предмет | <ol style="list-style-type: none"> 1. Відкривається головне меню гностичної гри з наступними можливими діями: обрати тип сцени, почати гру та вихід із гри 2. Відкривається ігрова сцена «Сад» з різноманіттям предметів та силуетами шуканих предметів на нижній панелі. 3. Знайдений предмет зникає зі списку силуетів та з ігрової області 4. З'являється екран успішного завершення гри з наступними можливими діями: почати спочатку та вийти з гри 5. Відкривається головне меню гностичної гри 6. Відкривається ігрова сцена «Візок» з різноманіттям предметів та зображеннями шуканих предметів на нижній панелі 7. Знайдений предмет зникає зі списку силуетів та з ігрової області 8. З'являється екран успішного завершення гри з наступними можливими діями: почати спочатку та вийти з гри |
| 5 | Натиснути на кнопку «Вихід» | Програмне забезпечення завершує свою роботу. |

4.4. Рекомендації щодо подальшого вдосконалення

Основною рекомендацією щодо вдосконалення розробленої системи є збір фокус-групи зі школярів молодших класів загальноосвітньої школи різного віку та отримання їх відгуків щодо комфортного користування програмним забезпеченням.

Також для покращення роботи системи та виключення затримки, яка присутня в соціально-психологічному модулі через збереження тимчасового аудіо файлу, завантаження його на хмарний сервіс для аналізу та очікування відповіді, буде доцільно розробити свою систему розпізнавання голосу, оскільки зараз, виходячи з аналізу існуючих технологій розпізнавання голосу, Google є єдиним, хто підтримує українську та російську мови.

Це до того ж, значно розширить цільову аудиторію та додасть функціональності наявність локалізації, бо в такому випадку дане програмне забезпечення можна буде використовувати ще й для тренування іншої мови або дітям, у кого інша рідна мова.

ВИСНОВКИ

Результатом виконання даного дипломного проекту є розроблений комп'ютерний додаток, який є ігровим програмним забезпеченням, що включає в себе не тільки поліпшені аналоги існуючих ігор, що виконують гностичну і інструментальну функції, але й ігри, які виконують соціально-психологічну функцію, зокрема, націлені на навчання базової ввічливості (використовується розпізнавання голосу для перевірки правильності відповіді).

Програма є достатньо легковісною, щоб батьки або вчителі мали змогу її запустити на майже будь-якому комп'ютері або ноутбукі. Єдине існуюче обмеження – операційна система має бути тільки Windows, але у майбутньому цей недолік буде виправлений, оскільки додаток реалізується на основі багатоплатформового інструменту для розроблення двовимірних та тривимірних додатків Unity.

Також система є легко розширюваною, оскільки кожна з ігор, що входить до складу програми, являє собою окрему ігрову сцену, що робить взаємозв'язок між іграми мінімальним. Ігрові сцени містять об'єкти, які будуть використовуватись у грі і можуть служити як для створення меню гри, так і для моделювання ігрового рівня.

Найбільш складним для реалізації компонентом програмного забезпечення був той, що включає в себе комп'ютерну гру, націлену на розвиток соціально-психологічного розвиваючого процесу, оскільки у ньому необхідно надати альтернативу живому спілкуванню учня і вчителя.

Саме цьому для реалізації гри такого типу ведеться робота з прикладним програмним інтерфейсом продукту з технологією розпізнавання мови, з назвою Google Speech Recognition API, який може забезпечити безперервне розпізнавання голосового потоку. Був обраним саме цей продукт, оскільки дуже важливою є точність розпізнавання мови дитини, а Google є однією з найбільш відомих компаній, яка має багато

даних для тренування своєї нейронної мережі та великі обчислювальні потужності для швидкого реагування на запити до API.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

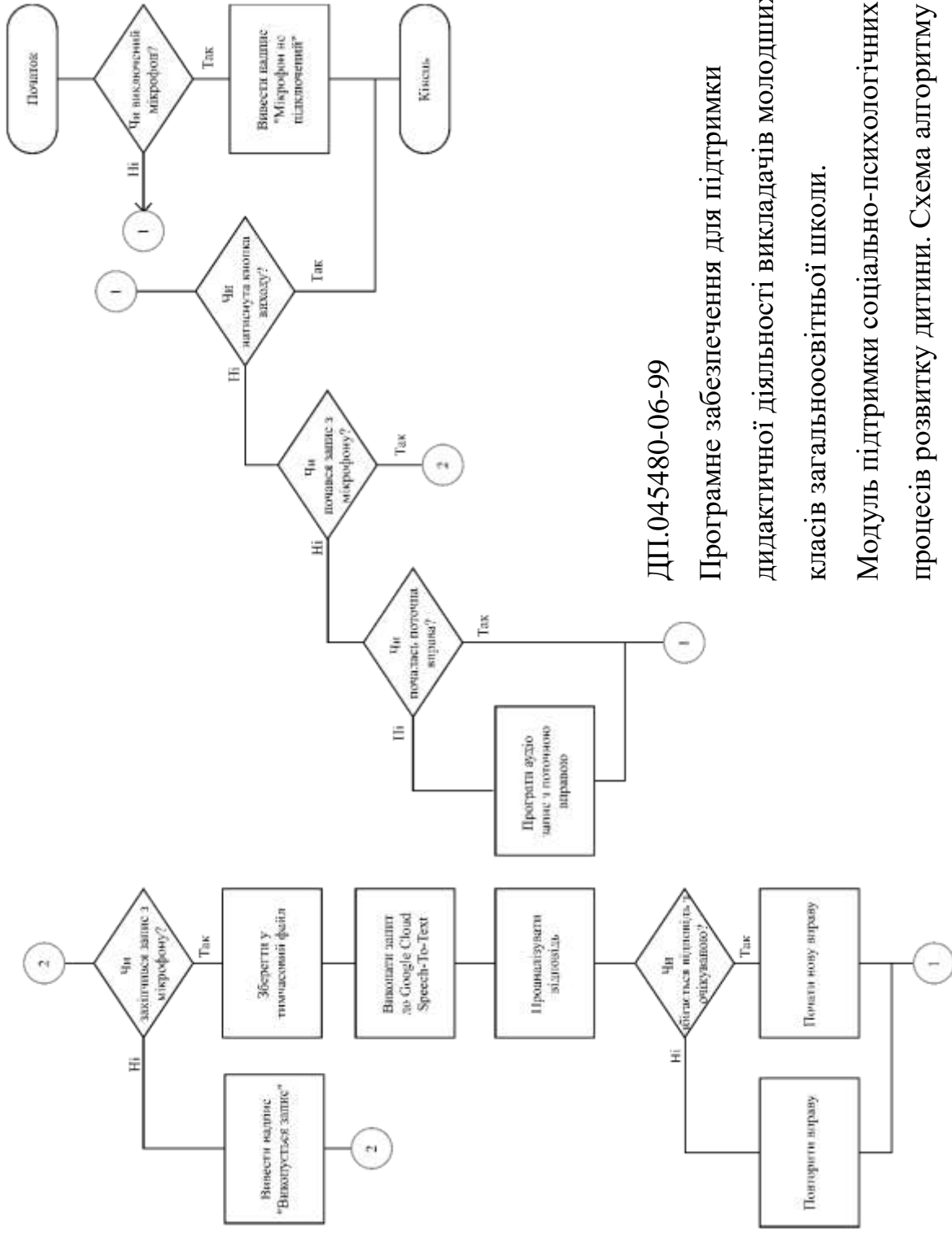
1. Копытина Т.Н. Обучающие игры: их функции, особенности и основные виды. [Электронный ресурс] – Режим доступа до ресурсу: <http://www.pws-conf.ru/nauchnaya/lss-2007/360-edagogicheskoe-soprovozhdenie-socializacii-lichnosti/8024-obuchayushchie-igry-ih-funkcii-osobennosti-i-osnovnye-vidy.html>.
2. Комп'ютерна гра "Учимся читать. Игры для обучения чтению" [Электронный ресурс] – Режим доступа до ресурсу: http://adalin.mospsy.ru/disc_r.shtml.
3. Комп'ютерна гра "Дракоша и 7 чудес света" [Электронный ресурс] – Режим доступа до ресурсу: https://www.logopedplus.ru/shop/catalog/znakomstvo_s_okruzhayushchim_mirom_tri/item3860/.
4. Комп'ютерна гра "Веселые игры для развития речи и слуха" [Электронный ресурс] – Режим доступа до ресурсу: <https://smallgames.ws/11165-veselye-igry-dlya-razvitiya-rechi-i-sluxa.html>.
5. Комп'ютерна гра "Губка Боб Квадратные Штаны учит печатать" [Электронный ресурс] – Режим доступа до ресурсу: <https://smallgames.ws/10611-gubka-bob-kvadratnye-shtany-uchit-pechatat.html>.
6. Интерактивный стол Kids Table 32 [Электронный ресурс] – Режим доступа до ресурсу: <http://hiteklab.vl.ru/services/21>.
7. Unity Manual. Working in Unity. [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.unity3d.com/Manual/UnityOverview.html>.
8. DirectX [Электронный ресурс] – Режим доступа до ресурсу: <https://ru.wikipedia.org/wiki/DirectX>.
9. OpenGL [Электронный ресурс] – Режим доступа до ресурсу: <https://ru.wikipedia.org/wiki/OpenGL>.

10. C# [Электронный ресурс] – Режим доступа до ресурсу:
https://ru.wikipedia.org/wiki/C_Sharp.
11. JavaScript [Электронный ресурс] – Режим доступа до ресурсу:
<https://ru.wikipedia.org/wiki/JavaScript>.
12. CryEngine [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.cryengine.com/>.
13. Шейдер [Электронный ресурс] – Режим доступа до ресурсу:
<https://ru.wikipedia.org/wiki/%D0%A8%D0%B5%D0%B9%D0%B4%D0%B5%D1%80>.
14. C++ [Электронный ресурс] – Режим доступа до ресурсу:
<https://ru.wikipedia.org/wiki/C%2B%2B>.
15. Unreal Engine [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.unrealengine.com/en-US/>.
16. Cocos2D-x [Электронный ресурс] – Режим доступа до ресурсу:
<https://ru.wikipedia.org/wiki/Cocos2d>.
17. Amazon Transcribe [Электронный ресурс] – Режим доступа до ресурсу:
<https://aws.amazon.com/ru/transcribe/faqs/#>.
18. Google Cloud Speech-to-Text [Электронный ресурс] – Режим доступа до ресурсу:
<https://cloud.google.com/speech-to-text/>.
19. Microsoft Azure Speech to Text [Электронный ресурс] – Режим доступа до ресурсу:
<https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/>.
20. Speech SDK [Электронный ресурс] – Режим доступа до ресурсу:
<https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/speech-sdk>.
21. IDEF0 [Электронный ресурс] – Режим доступа до ресурсу:
<https://ru.wikipedia.org/wiki/IDEF0>.
22. Ігрові активи [Электронный ресурс] – Режим доступа до ресурсу:
https://ru.wikipedia.org/wiki/%D0%98%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B0%D1%81%D1%81%D0%B5%D1%82.

23. Компонентно-орієнтоване програмування [Електронний ресурс] – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Component-based_software_engineering.
24. Audio Source [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.unity3d.com/ru/current/Manual/class-AudioSource.html>.
25. Microphone in Unity [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.unity3d.com/ScriptReference/Microphone.html>.

ДОДАТКИ

Додаток 1
Копії графічних матеріалів

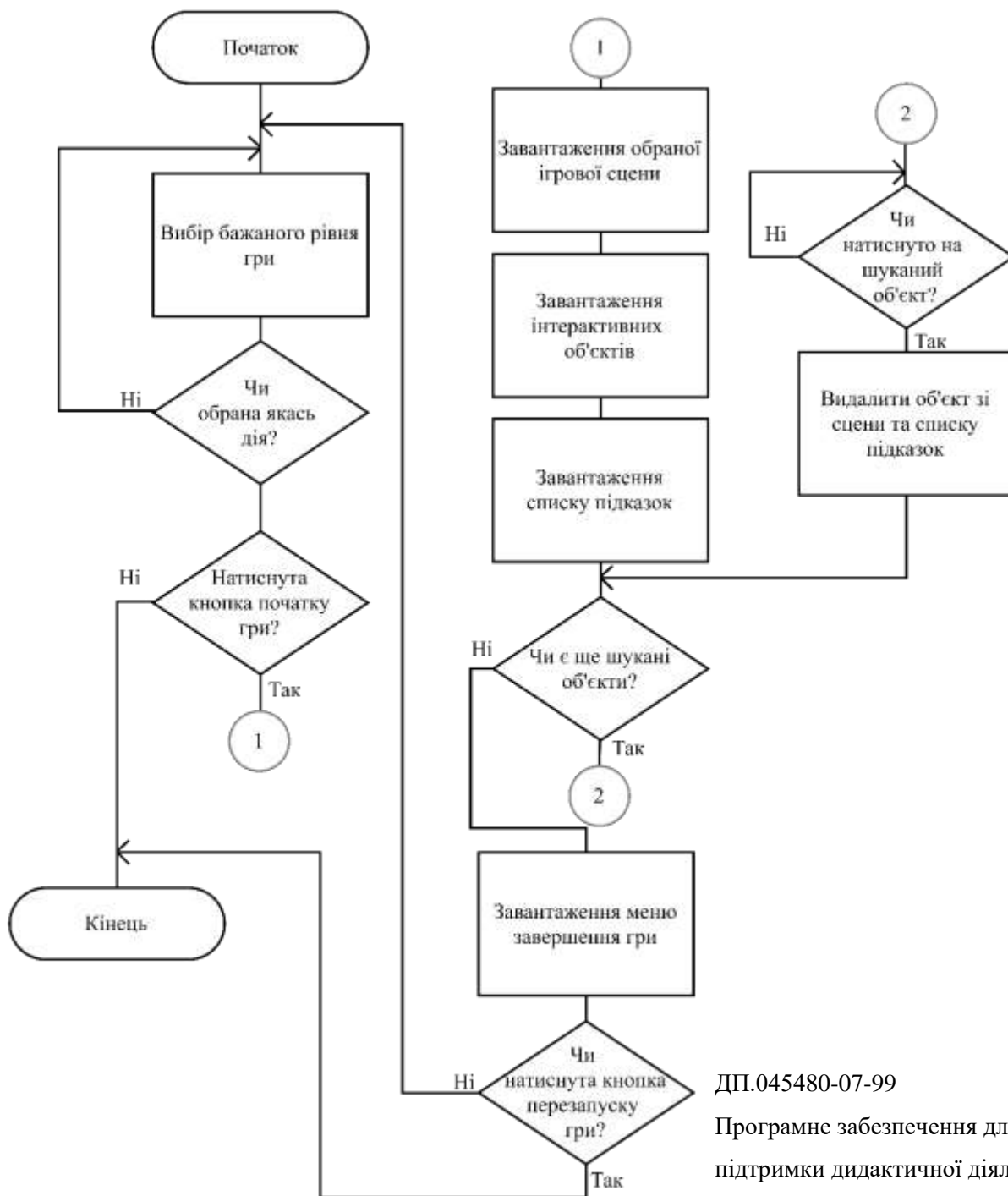


ДП.045480-06-99

Програмне забезпечення для підтримки

дидактичної діяльності викладачів молодших класів загальноосвітньої школи.

Модуль підтримки соціально-психологічних процесів розвитку дитини. Схема алгоритму



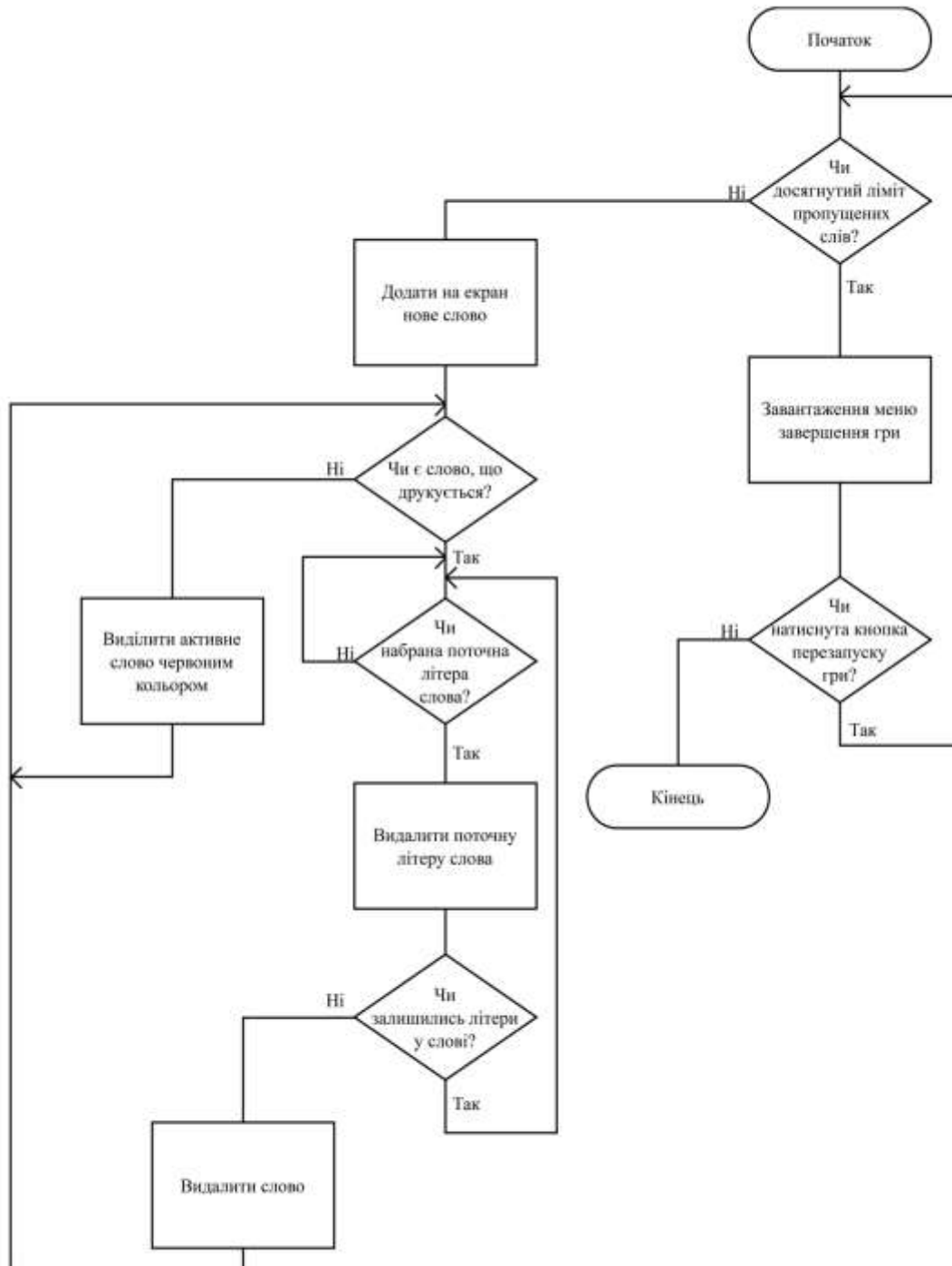
ДП.045480-07-99

Програмне забезпечення для підтримки дидактичної діяльності викладачів молодших класів загальноосвітньої школи.

Модуль підтримки гностичних процесів

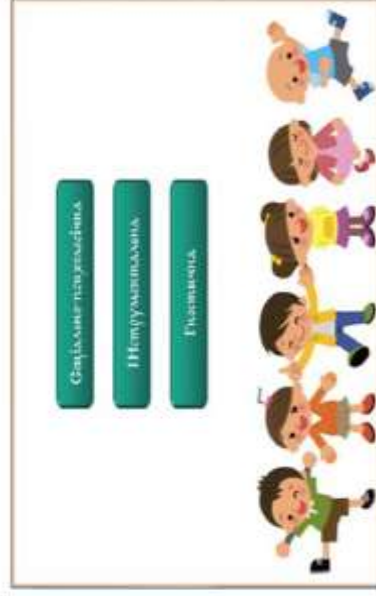
розвитку дитини. Схема алгоритму

СХЕМА АЛГОРИТМУ РОБОТИ МОДУЛЮ ПІДТРИМКИ
ІНСТРУМЕНТАЛЬНИХ ПРОЦЕСІВ РОЗВИТКУ ДИТИНИ



Левчук О.С., група КП-52

ІНТЕРФЕЙС ПРОГРАМНИХ МОДУЛІВ



Головне меню



Модуль, що впливає на інструментальні процеси розвитку дитини



Модуль, що впливає на соціально-психологічні процеси розвитку дитини



Модуль, що впливає на гностичні процеси розвитку дитини



Додаток 2

Лістинг модулю підтримки соціально-психологічних процесів розвитку дитини

```

[RequireComponent(typeof(AudioSource)) ]

public class GoogleVoiceSpeech : MonoBehaviour
{

    public GUIText TextBox;

    public List<AudioSource> audioLessons;
    public List<string> answers;
    [TextArea]
    public List<string> textToShow;
    public GUIText lessonBox;
    public List<Sprite> images;
    public GameObject forImage;

    struct ClipData
    {
        public int samples;
    }

    const int HEADER_SIZE = 44;

    private int minFreq;
    private int maxFreq;

    private bool micConnected = false;

    //A handle to the attached AudioSource
    private AudioSource goAudioSource;

    string apiKey;
    public int index = 0;
    bool audiofinished = false;
    public GameObject correct;
    public GameObject repeat;

    // Use this for initialization
    void Start()
    {

```



```

index = UnityEngine.Random.Range(0, answers.Count);

if (Microphone.devices.Length <= 0)
{
    Debug.LogWarning("Microphone not connected!");
}
else
{
    //Set 'micConnected' to true
    micConnected = true;

    //Get the default microphone recording capabilities
    Microphone.GetDeviceCaps(null, out minFreq, out maxFreq);

    if (minFreq == 0 && maxFreq == 0)
    {
        maxFreq = 44100;
    }
    goAudioSource = this.GetComponent<AudioSource>();
}

IEnumerator wait()
{
    yield return new WaitForSeconds(1f);
    correct.GetComponent<Canvas>().enabled = false;
    repeat.GetComponent<Canvas>().enabled = false;
}

void OnGUI()
{
    if (micConnected)
    {
        if (!Microphone.IsRecording(null))
        {
            StartCoroutine(wait());
        }
    }
}

```

```

if (!audiofinished)
{
    audiofinished = true;
    audioLessons[index].Play();
    lessonBox.text = textToShow[index];
    forImage.GetComponent<SpriteRenderer>().sprite = images[index];
}

//Case the 'Record' button gets pressed
if (GUI.Button(new Rect(Screen.width / 2 - 100, Screen.height - 75, 200, 50),
"Записати відповідь!") && audiofinished)
{
    goAudioSource.clip = Microphone.Start(null, true, 7, maxFreq);
}
}
else
{
    if (GUI.Button(new Rect(Screen.width / 2 - 100, Screen.height - 75, 200, 50),
"Перевірити!"))
    {
        float filenameRand = UnityEngine.Random.Range(0.0f, 10.0f);

        string filename = "testing" + filenameRand;

        Microphone.End(null); //Stop the audio recording

        Debug.Log("Recording Stopped");

        if (!filename.ToLower().EndsWith(".wav"))
        {
            filename += ".wav";
        }

        var filePath = Path.Combine("testing/", filename);
        filePath = Path.Combine(Application.persistentDataPath, filePath);
        Debug.Log("Created filepath string: " + filePath);

        Directory.CreateDirectory(Path.GetDirectoryName(filePath));
    }
}

```

```

SavWav.Save(filePath, goAudioSource.clip); //Save a temporary Wav File
Debug.Log("Saving @ " + filePath);

string apiURL =
"https://speech.googleapis.com/v1/speech:recognize?key=AIzaSyARop-2L9sG-
vOyFSDTHqctTfijfKucyV8";
string Response;

Debug.Log("Uploading " + filePath);
Response = HttpUploadFile(apiURL, filePath, "file", "audio/wav;
rate=44100");
Debug.Log("Response String: " + Response);

var jsonresponse = SimpleJSON.JSON.Parse(Response);

if (jsonresponse != null)
{
    string resultString = jsonresponse["results"][0].ToString();
    if (resultString != "")
    {
        var jsonResults = SimpleJSON.JSON.Parse(resultString);

        if (jsonResults != null)
        {
            string transcripts =
jsonResults["alternatives"][0]["transcript"].ToString().Trim('\\"').ToLower();
            Debug.Log("transcript string: " + transcripts);
            if (transcripts.Equals(answers[index].ToLower()))
            {
                TextBox.text = transcripts + " - Правильно!";
                correct.GetComponent<Canvas>().enabled = true;
                int oldIndex = index;
                while (index == oldIndex)
                    index = UnityEngine.Random.Range(0, answers.Count);
            }
            else
            {
                repeat.GetComponent<Canvas>().enabled = true;
                TextBox.text = transcripts + " - Неправильно :C";
            }
        }
    }
}

```

```

        audiofinished = false;

        }

    }

    File.Delete(filePath); //Delete the Temporary Wav file

}

GUI.Label(new Rect(Screen.width / 2 - 100, Screen.height - 100, 200, 50),
"Записую...");
}
}
else
{
    GUI.contentColor = Color.red;
    GUI.Label(new Rect(Screen.width / 2 - 100, Screen.height / 2 - 25, 200, 50),
"Мікрофон не підключений!");
}
}

public string HttpUploadFile(string url, string file, string paramName,
string contentType)
{

    System.Net.ServicePointManager.ServerCertificateValidationCallback += (o,
certificate, chain, errors) => true;
    Debug.Log(string.Format("Uploading {0} to {1}", file, url));

    Byte[] bytes = File.ReadAllBytes(file);
    String file64 = Convert.ToBase64String(bytes,
Base64FormattingOptions.None);

    Debug.Log(file64);

    try
    {

        var httpWebRequest = (HttpWebRequest)WebRequest.Create(url);
        httpWebRequest.ContentType = "application/json";
    }
}

```

```

httpWebRequest.Method = "POST";

using (var streamWriter = new
StreamWriter(httpWebRequest.GetRequestStream()))
{
    string json = "{ \"config\": { \"languageCode\" : \"ru-RU\" }, \"audio\" : {
\"content\" : \"\" + file64 + \"\"}}";

    Debug.Log(json);
    streamWriter.Write(json);
    streamWriter.Flush();
    streamWriter.Close();
}

var httpResponse = (HttpWebResponse)httpWebRequest.GetResponse();
Debug.Log(httpResponse);

using (var streamReader = new
StreamReader(httpResponse.GetResponseStream()))
{
    var result = streamReader.ReadToEnd();
    Debug.Log("Response:" + result);
    return result;
}

}

catch (WebException ex)
{
    var resp = new StreamReader(ex.Response.GetResponseStream()).ReadToEnd();
    Debug.Log(resp);
}

return "empty";

}

}

```

```

namespace SimpleJSON
{
    public enum JSONBinaryTag
    {
        Array            = 1,
        Class            = 2,
        Value            = 3,
        IntValue         = 4,
        DoubleValue      = 5,
        BoolValue        = 6,
        FloatValue       = 7,
    }

    public class JSONNode
    {
        #region common interface
        public virtual void Add(string aKey, JSONNode aItem){ }
        public virtual JSONNode this[int aIndex]    { get { return null; } }
set { } }
        public virtual JSONNode this[string aKey]    { get { return null; } }
set { } }
        public virtual string Value                  { get { return ""; } }
set { } }
        public virtual int Count                     { get { return 0; } }

        public virtual IEnumerable <string> Keys { get { yield
break; }}

        public virtual void Add(JSONNode aItem)
        {
            Add("", aItem);
        }

        public virtual JSONNode Remove(string aKey) { return null; }
        public virtual JSONNode Remove(int aIndex) { return null; }
        public virtual JSONNode Remove(JSONNode aNode) { return aNode; }

        public virtual IEnumerable<JSONNode> Childs { get { yield break; } }
        public IEnumerable<JSONNode> DeepChilds
        {

```

```

        get
        {
            foreach (var C in Childs)
                foreach (var D in C.DeepChilds)
                    yield return D;
        }
    }

    public override string ToString()
    {
        return "JSONNode";
    }
    public virtual string ToString(string aPrefix)
    {
        return "JSONNode";
    }

    #endregion common interface

    #region typecasting properties
    public virtual int AsInt
    {
        get
        {
            int v = 0;
            if (int.TryParse(Value,out v))
                return v;
            return 0;
        }
        set
        {
            Value = value.ToString();
        }
    }
    public virtual float AsFloat
    {
        get
        {
            float v = 0.0f;
            if (float.TryParse(Value,out v))

```

```

        return v;
    return 0.0f;
}
set
{
    Value = value.ToString();
}
}
public virtual double AsDouble
{
    get
    {
        double v = 0.0;
        if (double.TryParse(Value, out v))
            return v;
        return 0.0;
    }
    set
    {
        Value = value.ToString();
    }
}
public virtual bool AsBool
{
    get
    {
        bool v = false;
        if (bool.TryParse(Value, out v))
            return v;
        return !string.IsNullOrEmpty(Value);
    }
    set
    {
        Value = (value)?"true":"false";
    }
}
public virtual JSONArray AsArray
{
    get
    {

```



```

        return this as JSONArray;
    }
}
public virtual JSONClass AsObject
{
    get
    {
        return this as JSONClass;
    }
}

#endregion typecasting properties

#region operators
public static implicit operator JSONNode(string s)
{
    return new JSONData(s);
}
public static implicit operator string(JSONNode d)
{
    return (d == null)?null:d.Value;
}
public static bool operator ==(JSONNode a, object b)
{
    if (b == null && a is JSONLazyCreator)
        return true;
    return System.Object.ReferenceEquals(a,b);
}

public static bool operator !=(JSONNode a, object b)
{
    return !(a == b);
}
public override bool Equals (object obj)
{
    return System.Object.ReferenceEquals(this, obj);
}
public override int GetHashCode ()
{

```

```

        return base.GetHashCode();
    }

#endregion operators

internal static string Escape(string aText)
{
    string result = "";
    foreach(char c in aText)
    {
        switch(c)
        {
            case '\\': result += "\\\\"; break;
            case '\"': result += "\\\""; break;
            case '\n': result += "\\n" ; break;
            case '\r': result += "\\r" ; break;
            case '\t': result += "\\t" ; break;
            case '\b': result += "\\b" ; break;
            case '\f': result += "\\f" ; break;
            default   : result += c      ; break;
        }
    }
    return result;
}

public static JSONNode Parse(string aJSON)
{
    Stack<JSONNode> stack = new Stack<JSONNode>();
    JSONNode ctx = null;
    int i = 0;
    string Token = "";
    string TokenName = "";
    bool QuoteMode = false;
    while (i < aJSON.Length)
    {
        switch (aJSON[i])
        {
            case '{':
                if (QuoteMode)

```

```

{
    Token += aJSON[i];
    break;
}
stack.Push(new JSONClass());
if (ctx != null)
{
    TokenName = TokenName.Trim();
    if (ctx is JSONArray)
        ctx.Add(stack.Peek());
    else if (TokenName != "")
        ctx.Add(TokenName, stack.Peek());
}
TokenName = "";
Token = "";
ctx = stack.Peek();
break;

case '[':
    if (QuoteMode)
    {
        Token += aJSON[i];
        break;
    }

    stack.Push(new JSONArray());
    if (ctx != null)
    {
        TokenName = TokenName.Trim();
        if (ctx is JSONArray)
            ctx.Add(stack.Peek());
        else if (TokenName != "")
            ctx.Add(TokenName, stack.Peek());
    }
    TokenName = "";
    Token = "";
    ctx = stack.Peek();
break;

case '}':

```

```

        case ']':
            if (QuoteMode)
            {
                Token += aJSON[i];
                break;
            }
            if (stack.Count == 0)
                throw new Exception("JSON Parse: Too many
closing brackets");

            stack.Pop();
            if (Token != "")
            {
                TokenName = TokenName.Trim();
                if (ctx is JSONArray)
                    ctx.Add(Token);
                else if (TokenName != "")
                    ctx.Add(TokenName, Token);
            }
            TokenName = "";
            Token = "";
            if (stack.Count > 0)
                ctx = stack.Peek();
            break;

        case ':':
            if (QuoteMode)
            {
                Token += aJSON[i];
                break;
            }
            TokenName = Token;
            Token = "";
            break;

        case '"':
            QuoteMode ^= true;
            break;

        case ',':

```

```

        if (QuoteMode)
        {
            Token += aJSON[i];
            break;
        }
        if (Token != "")
        {
            if (ctx is JSONArray)
                ctx.Add(Token);
            else if (TokenName != "")
                ctx.Add(TokenName, Token);
        }
        TokenName = "";
        Token = "";
    break;

    case '\r':
    case '\n':
    break;

    case ' ':
    case '\t':
        if (QuoteMode)
            Token += aJSON[i];
    break;

    case '\\':
        ++i;
        if (QuoteMode)
        {
            char C = aJSON[i];
            switch (C)
            {
                case 't' : Token += '\t'; break;
                case 'r' : Token += '\r'; break;
                case 'n' : Token += '\n'; break;
                case 'b' : Token += '\b'; break;
                case 'f' : Token += '\f'; break;
                case 'u':
            {

```

```

                string s = aJSON.Substring(i+1,4);
                Token += (char)int.Parse(s,
System.Globalization.NumberStyles.AllowHexSpecifier);
                i += 4;
                break;
            }
            default : Token += C; break;
        }
    }
    break;

    default:
        Token += aJSON[i];
        break;
    }
    ++i;
}
if (QuoteMode)
{
    throw new Exception("JSON Parse: Quotation marks seems to be
messed up.");
}
return ctx;
}

public virtual void Serialize(System.IO.BinaryWriter aWriter) {}

public void SaveToStream(System.IO.Stream aData)
{
    var W = new System.IO.BinaryWriter(aData);
    Serialize(W);
}

#if USE_SharpZipLib
public void SaveToCompressedStream(System.IO.Stream aData)
{
    using (var gzipOut = new
ICSharpCode.SharpZipLib.BZip2.BZip2OutputStream(aData))
    {
        gzipOut.IsStreamOwner = false;
    }
}

```

```

        SaveToStream(gzipOut);
        gzipOut.Close();
    }
}

public void SaveToCompressedFile(string aFileName)
{
    #if USE_FileIO
        System.IO.Directory.CreateDirectory((new
System.IO.FileInfo(aFileName)).Directory.FullName);
        using(var F = System.IO.File.OpenWrite(aFileName))
        {
            SaveToCompressedStream(F);
        }
    #else
        throw new Exception("Can't use File IO stuff in webplayer");
    #endif
}

public string SaveToCompressedBase64()
{
    using (var stream = new System.IO.MemoryStream())
    {
        SaveToCompressedStream(stream);
        stream.Position = 0;
        return System.Convert.ToBase64String(stream.ToArray());
    }
}

```

Додаток 3
Копія презентації

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО”



ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ ДИДАКТИЧНОЇ ДІЯЛЬНОСТІ ВИКЛАДАЧІВ МОЛОДШИХ КЛАСІВ ЗАГАЛЬНООСВІТНЬОЇ ШКОЛИ

Виконала: Левчук Ольга Сергіївна

Науковий керівник: к.т.н., доцент Заболотня Тетяна Миколаївна

Київ – 2019



АКТУАЛЬНІСТЬ

Категорії розвиваючих процесів:

- інструментальна;
- гностична;
- соціально-психологічна.



АКТУАЛЬНІСТЬ

Проблеми існуючих програмних рішень:

- відсутність універсального програмного забезпечення, яке включало б вправи для впливу на розвиток учнів молодших класів в декількох категоріях розвиваючих процесів;
- потреба купівлі додаткового обладнання.

ІСНУЮЧІ ПРОГРАМНІ РІШЕННЯ





ПОСТАНОВКА ЗАДАЧІ

Мета проекту: забезпечення підтримки розвитку дитини з точки зору всіх існуючих категорій розвиваючих процесів в ігровій формі.

Завдання: розробити програмне забезпечення у формі ігрового комплексу для розвитку дитини в інструментальному, гностичному та соціально-психологічному напрямках.

Вимоги до програмного забезпечення

- вимоги, сформовані UI та UX дизайнерами;
- вимоги до організації навчального процесу у ігровій формі, сформовані авторами досліджених педагогічних статей;
 - створення ігрової атмосфери;
 - проведення гри, в результаті якої повинна бути виконана поставлена задача;
 - підведення підсумків.
- вимоги до структури організації сцен в обраному ігровому русії.



ЗАСОБИ РЕАЛІЗАЦІЇ

Ігрові рушії



ЗАСОБИ РЕАЛІЗАЦІЇ

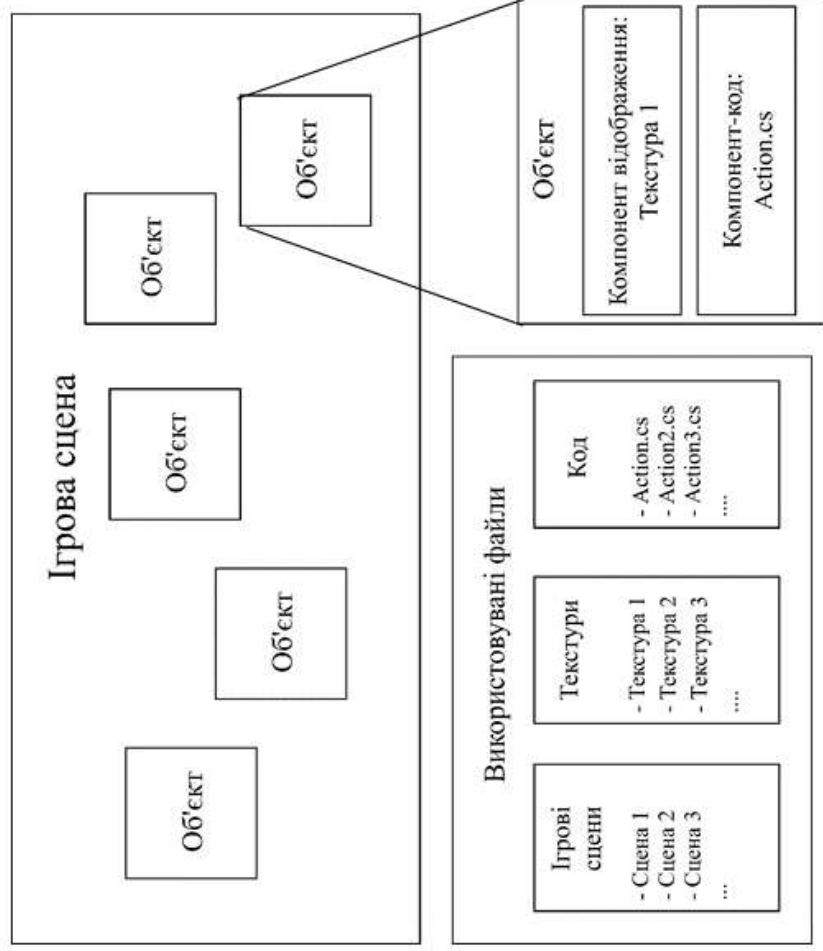


API для використання технології розпізнавання мови

amazon Google



СТРУКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Модуль, що автоматизує соціально-психологічну вправу

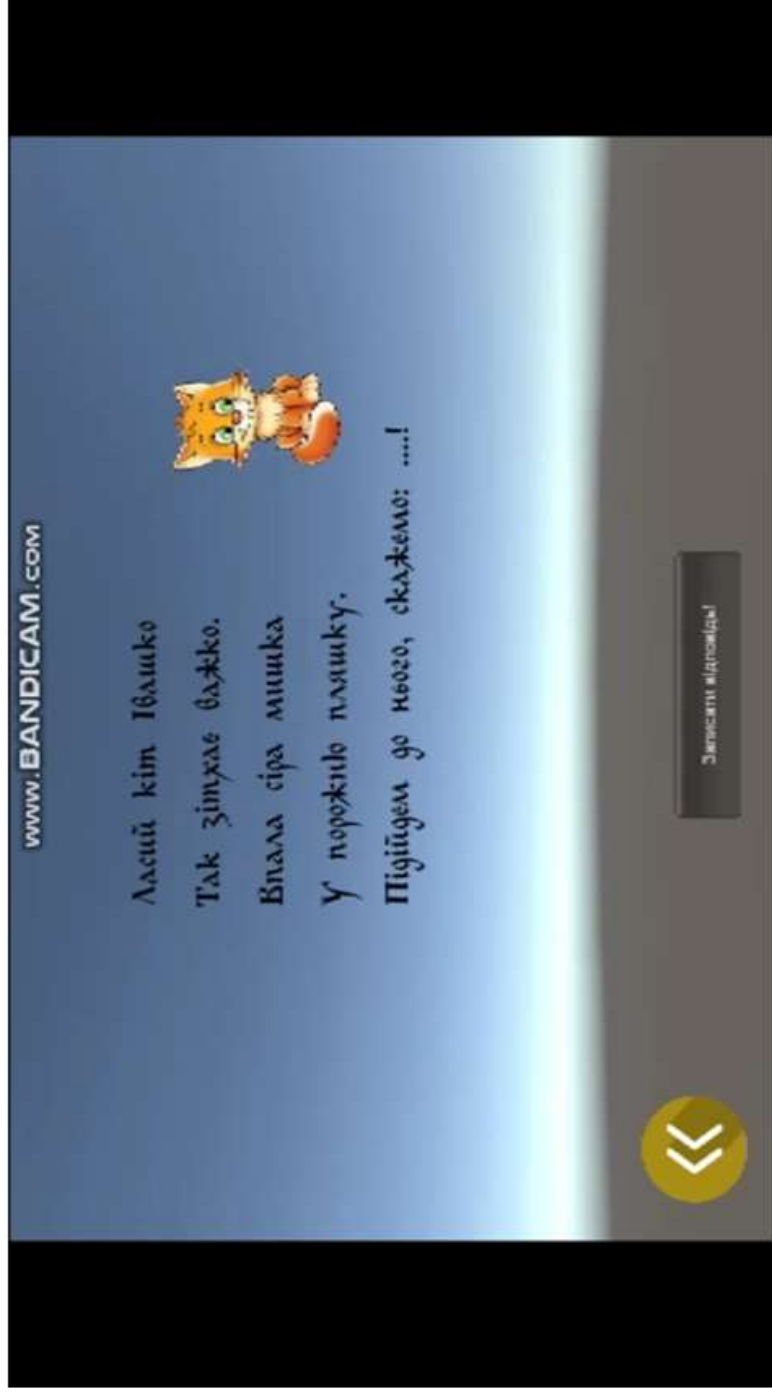




Схема алгоритму роботи модулю

Модуль, що автоматизує гностичну вправу



www.BANDICAM.com

Заховані речі

Обери рівень

Сад ▾



Вихід

Початок



Схема алгоритму роботи модулю



Модуль, що автоматизує інструментальну вправу





Схема алгоритму роботи модулю



ПУБЛІКАЦІЇ

Апробація роботи була проведена на IV Міжнародній науково-технічній конференції «Поліграфічні, мультимедійні та web-технології» РММ 2019 (Харків, 14-17 травня) та опубліковані як тези доповідей.



ВИСНОВКИ

- Результатом виконання даного дипломного проекту є розроблений комп'ютерний додаток, який містить:
- соціально-психологічну складову (модуль, націлений на вивчення базової ввічливості);
 - гностичну складову (модуль, націлений на пошук предметів);
 - інструментальну складову (модуль, націлений на навчання швидкому друку).



Дякую за увагу!

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Науковий керівник кафедри

_____ І.А. Дичка

“ ____ ” _____ 2018 р.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ ДИДАКТИЧНОЇ
ДІЯЛЬНОСТІ ВИКЛАДАЧІВ МОЛОДШИХ КЛАСІВ
ЗАГАЛЬНООСВІТНЬОЇ ШКОЛИ
Програма та методика тестування
ДП.045480-04-51

“ПОГОДЖЕНО”

Керівник проекту:

_____ Т. М. Заболотня

Нормоконтроль:

_____ М.В. Онай

Виконавець:

_____ О. С. Левчук

ЗМІСТ

| | |
|--------------------------------------|---|
| 1. Об'єкт випробувань | 3 |
| 2. Мета тестування | 3 |
| 3. Методи тестування..... | 3 |
| 4. Засоби та порядок тестування..... | 4 |

1. ОБ'ЄКТ ВИПРОБУВАНЬ

Програмне забезпечення для підтримки дидактичної діяльності викладачів молодших класів загальноосвітньої школи.

2. МЕТА ТЕСТУВАННЯ

У процесі тестування має бути перевірено наступне:

- 1) функціональна працездатність кожного програмного модулю;
- 2) наявність доступу до технології розпізнавання мови Google Cloud Speech-to-Text;
- б) функціональність та доступність кожного інтерактивного елементу сцен;
- 7) відповідність дизайну програмного забезпечення вимогам Технічного завдання.

3. МЕТОДИ ТЕСТУВАННЯ

Для впевненості, що програмне забезпечення задовольняє всім поставленим вимогам та функціонує правильно, було використано наступні методи тестування:

- 1) розроблено набір тест-кейсів димового тестування;
- 2) проведено функціональне ручне тестування (Manual testing) для перевірки функціональних вимог;
- 3) стрес тестування, для визначення стійкості системи.

Димовий тест – це короткі цикли тестів, які проводяться під дуже низьким навантаженням. Цей вид продуктивного тестування підкреслює, що програма працює як очікується. Цей термін походить від апаратного тестування, де, якщо дим генерується (буквально), це означає, що тест не вдався, і більше тестування не потрібне.

4. ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Працездатність веб-додатку перевіряється шляхом:

- 1) виконання сценарію димового тестування після додавання нового функціоналу;
- 2) ручне тестування кожного програмного модулю на відповідність функціональним вимогам;
- 3) ручне тестування роботи с технологією розпізнавання мови на відповідність функціональним вимогам;
- 4) стрес-тестування кожного програмного модулю шляхом навантаження, що значно перевищують очікувані;
- 5) тестування відповідності типу даних, отриманих у якості відповіді від технології розпізнавання мови, з типом даних, які оброблюються програмним додатком;
- 6) тестування функціональності та доступності кожного інтерактивного елементу сцен.

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Науковий керівник кафедри

_____ І.А. Дичка

“ ____ ” _____ 2019 р.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ ДИДАКТИЧНОЇ
ДІЯЛЬНОСТІ ВИКЛАДАЧІВ МОЛОДШИХ КЛАСІВ
ЗАГАЛЬНООСВІТНЬОЇ ШКОЛИ

Керівництво користувача

ДП.045480-05-34

“ПОГОДЖЕНО”

Керівник проекту:

_____ Т. М. Заболотня

Нормоконтроль:

_____ М.В. Онай

Виконавець:

_____ О. С. Левчук

ЗМІСТ

| | |
|--|---|
| 1. Опис структури додатку | 3 |
| 2. Опис головного меню | 4 |
| 3. Опис програмного модулю, що впливає на соціально-психологічні процеси розвитку дитини | 5 |
| 4. Опис програмного модулю, що впливає на інструментальні процеси розвитку дитини | 6 |
| 5. Опис програмного модулю, що впливає на гностичні процеси розвитку дитини..... | 7 |

1. Опис структури додатку

Розроблюване програмне забезпечення складається з сукупності ігрових сцен, які пов'язані між собою сценою головного меню.

Складовою, що забезпечує підтримку соціально-психологічної функції програмного забезпечення, є гра на вивчення школярами правил базової ввічливості в ігровій формі за допомогою технології розпізнавання мови Google Cloud Speech-to-Text.

Складовою, що забезпечує підтримку інструментальної функції програмного забезпечення, є гра на навчання швидкому друку.

Складовою, що забезпечує підтримку гностичної функції програмного забезпечення, є гра на пошук предметів.

2. Опис головного меню

Головне меню містить наступні елементи (рис. 1):

- кнопка переходу до модулю, що впливає на соціально-психологічні процеси розвитку дитини;
- кнопка переходу до модулю, що впливає на гностичні процеси розвитку дитини;
- кнопка переходу до модулю, що впливає на гностичні процеси розвитку дитини.



Рис. 1. Інтерфейс головного меню

3. Опис програмного модулю, що впливає на соціально-психологічної процеси розвитку дитини

Програмний модуль, що впливає на соціально-психологічної процеси розвитку дитини, розроблений у вигляді окремої ігрової сцени і є єдиним модулем, який потребує підключення до Інтернет. Даний модуль містить наступні елементи (рис. 2)

- текст вправи;
- невелике зображення, яке асоціюється із вправою;
- кнопка для запису відповіді;
- кнопка для повернення до головного меню.

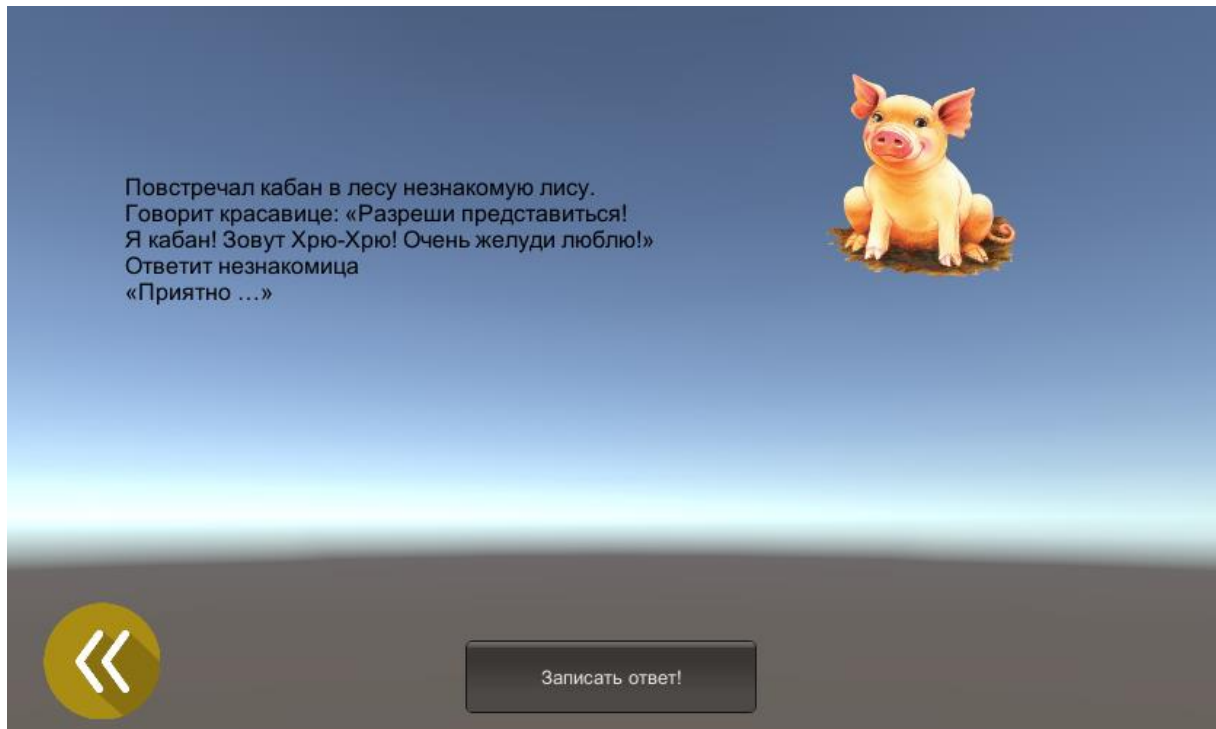


Рис. 2. Інтерфейс модулю, що впливає на соціально-психологічні процеси розвитку дитини

4. Опис програмного модулю, що впливає на інструментальні процеси розвитку дитини

Модуль, що впливає на інструментальні процеси розвитку дитини має бути сфокусований лише на вправі, оскільки за допомогою даного додатку мають формуватися основні навички та вміння, саме цьому більшу частину екрану в розробленому програмному модулі займають слова (рис. 3)

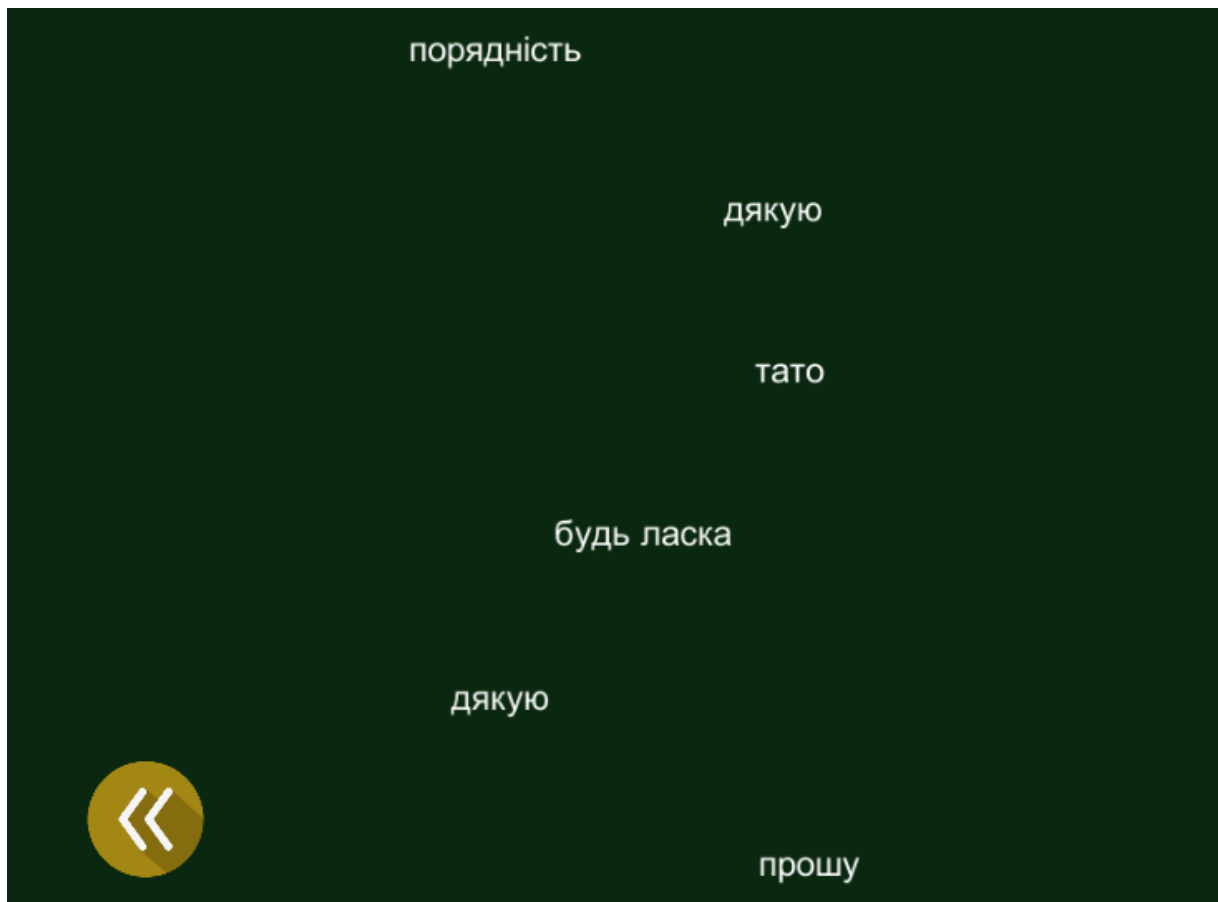


Рис. 3. Інтерфейс модулю, що впливає на інструментальні процеси розвитку дитини

5. Опис програмного модулю, що впливає на гностичні процеси розвитку дитини

Модуль, що впливає на гностичні процеси розвитку дитини, включає в себе декілька сцен, а саме сцену головного меню, кожну з двох ігрових сцен та сцену успішного завершення гри (рис. 4-7).

Сцена головного меню містить:

- випадające меню зі списком доступних рівнів гри: «Сад» або «Візок»;
- кнопки для виходу з гри та початку гри з обраним рівнем.

Сцена обраного ігрового рівня містить:

- панель з підказками, які предмети необхідно знайти;
- інтерактивні ігрові об'єкти, які зникають з екрану, якщо «силует» цього об'єкту мітиться на панелі підказок.

Сцена завершення гри містить кнопки виходу з гри та повторної спроби, яка повертає користувача на сцену головного меню.

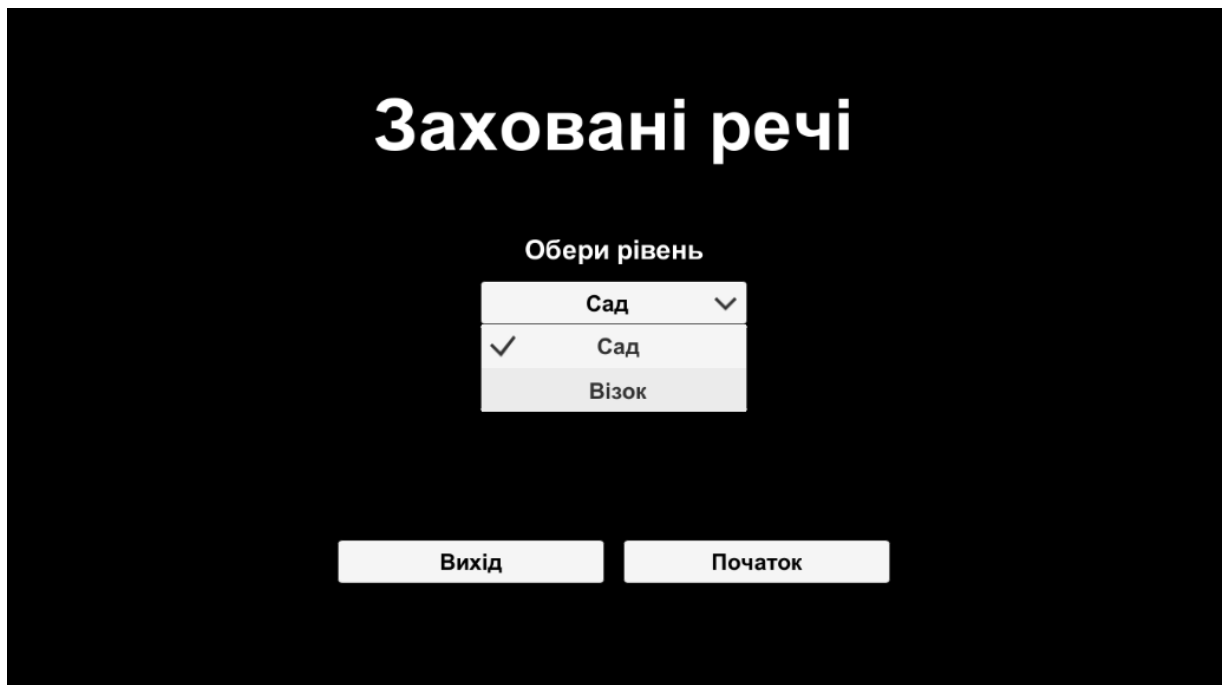


Рис. 4. Інтерфейс головного меню

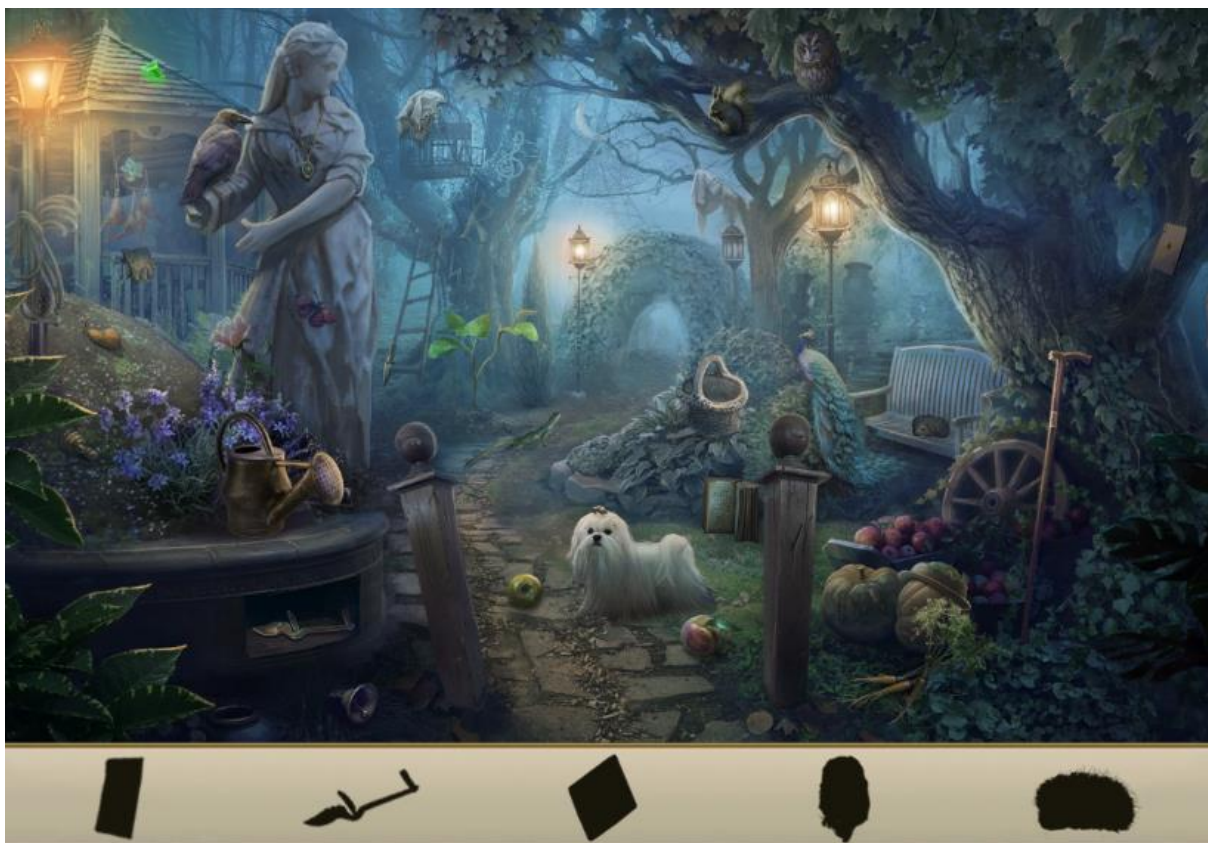


Рис 5. Інтерфейс ігрової сцени «Сад»



Рис. 6. Інтерфейс ігрової сцени «Візок»



Рис. 7. Інтерфейс екрану успішного завершення гри